

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2003-58408
(P2003-58408A)

(43) 公開日 平成15年2月28日 (2003.2.28)

(51) Int.Cl.⁷

G 0 6 F 12/00

識別記号

5 4 5

F I

G 0 6 F 12/00

テーマコード(参考)

5 4 5 B 5 B 0 8 2

5 4 5 A

審査請求 未請求 請求項の数 6 O L (全 34 頁)

(21) 出願番号 特願2001-245524(P2001-245524)

(22) 出願日 平成13年8月13日 (2001.8.13)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 茂木 和彦

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 加納 義樹

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 100093492

弁理士 鈴木 市郎 (外1名)

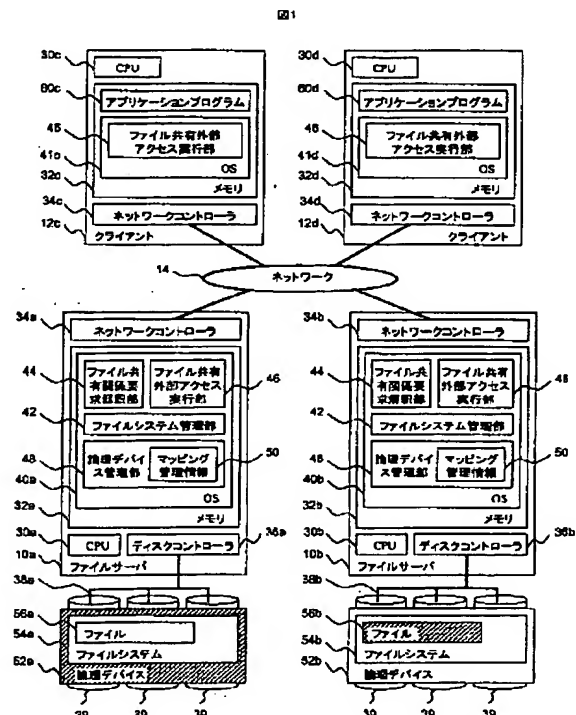
最終頁に続く

(54) 【発明の名称】 情報処理システム

(57) 【要約】

【課題】 ファイルサーバが提供する名前空間やファイル利用量・ファイル作成数の制限とファイル実体に対して等価な複数の名前を付加する機能を保持しながら、ファイルサーバが他ファイルサーバ上の領域を利用することを可能にする。

【解決手段】 ファイルサーバ10a上の複数の外部記憶装置39を1つの論理デバイスに見せる論理デバイス管理部48は、他のファイルサーバ10b上に割り当てられたファイル56bを外部記憶装置39と等価に扱うことにより、他ファイルサーバ10bの記憶領域を含めた領域にファイルシステム54aのデータを記憶する。また、ファイルシステム管理部48は、ファイルサーバ上のファイルシステム154a、154bを管理し、他ファイルシステムへの外部リンク機能とファイルシステム内を幾つかのディレクトリツリー毎に分割し、独立にファイル利用量・ファイル作成数管理を行う。



【特許請求の範囲】

【請求項 1】 ファイル管理のための手段であるファイルシステムをそれぞれが独立して保持し、それぞれ独立した名前空間によるファイル共有手段を有する記憶装置を備える情報処理システムにおいて、前記憶装置は、他の記憶装置が管理するファイルシステム内の領域を利用し、その記憶装置が提供する名前空間の記憶容量を増加させる手段を有することを特徴とする情報処理システム。

【請求項 2】 前記憶装置は、記憶装置が解釈可能なファイル共有処理要求を用いて他の記憶装置が管理するファイルシステム内の領域を利用する手段を有することを特徴とする請求項 1 記載の情報処理システム。

【請求項 3】 前記憶装置は、他の記憶装置が管理するファイルシステム内の領域を利用するために前記憶装置が保持する物理的記憶装置と他の記憶装置上の領域とを纏めて 1 つの論理的記憶装置として利用可能とする手段を有することを特徴とする請求項 1 または 2 記載の情報処理システム。

【請求項 4】 前記憶装置は、作成された論理的記憶装置上に前記憶装置が保持するファイルシステムのデータを記録する手段を有することを特徴とする請求項 3 記載の情報処理システム。

【請求項 5】 前記憶装置は、前記憶装置が保持するファイルシステム毎に独立にファイル利用量やファイル作成数に関する上限管理を行う手段と、他の記憶装置上のファイル利用量やファイル作成数に関する上限管理の管理データを更新する手段と、他の記憶装置上のファイル利用量やファイル作成数に関する上限管理の管理データの最適割り当て値を動的に計算する手段とを有することを特徴とする請求項 1 記載の情報処理システム。

【請求項 6】 前記憶装置が保持するファイルシステムは、他の記憶装置が保持するファイルシステム上のファイルやディレクトリの実体を指し示す名前を作成する手段と、他の記憶装置から指し示される記憶装置が保持するファイルシステム上のファイルやディレクトリの実体を管理する手段とを有することを特徴とする請求項 1 または 5 記載の情報処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、情報処理システムに係り、特に、ファイル共有プロトコルによりファイル共有機能を提供する記憶装置を有する情報処理システムに関する。

【0002】

【従来の技術】 複数のコンピュータ相互間でネットワークを介してファイルの共有を行う場合、一般に、NFS や CIFS のようなファイル共有プロトコルを用いて処理を実行している。このファイル共有処理は、ディレクトリやファイルに対する各種処理要求を出すクライアン

ト側と、ファイルのデータを記録・管理し、クライアントからの要求を処理するサーバ側とに分類することができる。この場合、特に、サーバ側の処理を行うコンピュータ記憶装置をファイルサーバと呼ぶ。

【0003】 ファイル共有処理においても、ファイルシステムで実現されている機能が提供されている。この機能は、例えば、ユーザやユーザの集合であるグループによりアクセス制限をかけたたり、また、特定の所有者や特定のグループにより所有されるファイルの利用量や作成ファイル数に上限をかけたたりする機能である。また、あるファイル実体に対し、完全に等価に扱うことができる複数の名前を付加する機能が存在する。

【0004】 ファイルサーバにおいて、共有されるファイルの名前空間やそのファイル実体の管理を行う従来技術として、「The Design and Implementation of the 4.4 BSD Operating System (Marshall Kirk McKusick 他著、ISBN 0-201-54979-4)」の Chapter9 に記載された技術が知られている。この従来技術の文献の中には、4.4 BSD Operating System における NFS 機能の実装について記述されている。この実装において、ファイルサーバは、ファイルサーバ内のローカルファイルシステム毎にファイル共有を行うか否かを決定し、ファイル共有を行うファイルシステムを外部に公開している。

【0005】 前述した文献の中の Chapter7、Chapter8 には、ローカルファイルシステムの実装について記述されている。ここで説明されているファイルシステムは、ファイル実体に対し、完全に等価に扱うことができる複数の名前を付加する機能を保持している。また、前述した文献の Chapter7 には、ローカルファイルシステム内で、特定の所有者や特定のグループにより所有されるファイルの利用量や作成ファイル数に上限をかけたたりする機能の実装方式について記述されている。

【0006】 また、文献「Using Samba (Robert Eckstein 他著、ISBN 1-56592-449-5)」には、CIFS プロトコルを用いたファイルサーバ機能を UNIX (登録商標) 互換オペレーティングシステム上で実現するプログラム *samba* において、それが実行されているサーバ内部で認識されている名前空間を元に、動作構成情報により指定されたディレクトリやアクセス認証時に指定されたユーザのホームディレクトリ以下のディレクトリツリーを外部に公開し、そこに存在するファイルの共有を行うことが解説されている。

【0007】 また、文献「NFS illustrated (Brent Callaghan 著、ISBN 0-201-32570-5)」の Chapter13 には、Andrew File System (AFS) や AFS をベースに開発された DCE/DFS と呼ばれる分散ファイルシステムに関する技術が説明されている。AFS は、cell と呼ばれる単位の中で名前空間の共有やサーバ機能の提供を行っている。そして、1 つの cell の中にデータを保持するサーバを複数存在させることができる。ファイルやディ

レクトリは、それらの集合として構成される volume と呼ばれる単位を用いて記録位置の管理が行われる。volume の記憶位置は、volume location database (VLDB) と呼ばれる管理情報により管理され、これの管理機能を提供するサーバが存在する。クライアントは、ファイルにアクセスする際、volume 識別子を含むファイル識別子を用いて、VLDB からアクセスするファイルを保持している volume がどこに記録されているかを判断してアクセスを行う。DCE/DFS でも同様の機構で管理される。

【0008】さらに、前述した文献「NFS illustrated」の Chapter13 には、Windows (登録商標) NT4.0 における Dfs 機能についての説明が記述されている。この機能は、独立に名前空間が管理されているファイルサーバをまとめて 1 つの名前空間としてアクセスすることを可能とするものである。この機能利用時の処理の概略は以下の通りである。すなわち、あるファイルサーバが、Dfs 機能により、そのファイルサーバから別のサーバへのリンク点を含むパスのオープン処理要求を受け取った際、その要求を受けたサーバはエラーを返す。このとき、クライアントは、そのパス名に対して、Dfs を管理するサーバに対して正しいアクセス先のサーバとそのアクセス先名を要求し、新しく取得したアクセス先に再度アクセスすることにより正しい処理を行うことができる。

【0009】

【発明が解決しようとする課題】前述した従来技術によるファイルサーバは、データの記憶容量に上限が存在する。ファイルサーバによっては記憶装置の増設が可能であるが、常に記憶装置の増設を行うことができるとは限らない。このため、あるファイルサーバに対する容量に関する要求が運用中にその上限値を超えた場合、その要求を満たすには、そのファイルサーバ上に存在するファイルを複数のファイルサーバに分割する必要が生じる。また、複数のファイルサーバを独立に運用している環境において、1 つのファイルサーバが容量不足になったものの他ファイルサーバに余裕が存在する場合、あるファイルサーバの容量不足を解消するためにファイルサーバ上のデータの一部を記憶容量に余裕のあるサーバに移動して利用することはシステムの効率的な利用の観点から好ましいことである。このとき、分割された後も 1 つのファイルサーバ上で管理されていたときと同様に、クライアントのアクセスやサーバ管理が行える機能を提供することが望ましい。また、複数のファイルサーバが存在する環境において、あたかも 1 つのファイルサーバで管理されているように記憶装置を用いることができれば、その分サーバ管理の負荷を低減することができる。

【0010】複数のファイルサーバを利用する場合、単一のファイルサーバ利用時とは異なる機構を用意する必要があり、かつ、存在意義の高い機能として、以下の 3 つのものを挙げることができる。1 つ目は、ファイルに

対して単一の名前空間を用いてアクセス可能とする機能である。2 つ目は、特定のユーザや特定のユーザの集合であるグループに対して、その名前空間内のファイル利用量やその作成ファイル数に上限値を設けることである。3 つ目は、ファイルの実体が同一のものであるが、それに対して完全に等価に扱うことができる複数の名前を制約なく名前空間内に保持する機能である。前述の 1 つめと 3 つめとの機能は利用者の立場から望まれるものであり、2 つめの機能は管理者の立場から望まれるものである。

【0011】前述した従来技術は、前記のように異なった名前空間として管理されているものをまとめて 1 つの名前空間として見せる機能を有している。しかし、異なった名前空間として管理されているものをまとめて 1 つの名前空間としてまとめた際に、ファイル利用量やファイル作成数、ファイル実体の管理を統一化する機能は提供されていない。

【0012】前述のように、従来技術は、独立して異なった名前空間を提供することができるファイルサーバが複数存在する環境で、ファイルサーバ毎の名前空間やファイル利用量やファイル作成数の制限とファイル実体に対して等価な複数の名前を付加する機能を保持しながらファイルサーバ間で記憶領域を融通することができないという問題点を有している。

【0013】本発明の目的は、複数のコンピュータ記憶装置が存在するときに、あるコンピュータ記憶装置が提供していた名前空間に所属するファイルやディレクトリの記憶に、他のコンピュータ記憶装置の領域を利用することができる情報処理システムを提供することにある。すなわち、本発明の目的は、ファイルサーバ毎の名前空間やファイル利用量やファイル作成数の制限とファイル実体に対して等価な複数の名前を付加する機能を保持しながら、あるファイルサーバが他ファイルサーバ上の記憶領域を利用することを可能にし、複数のファイルサーバが存在する環境における記憶装置の運用管理の効率化を図ることのできる情報処理システムを提供することにある。

【0014】

【課題を解決するための手段】本発明によれば前記目的は、ファイル管理のための手段であるファイルシステムをそれぞれが独立して保持し、それぞれ独立した名前空間によるファイル共有手段を有する記憶装置を備える情報処理システムにおいて、前記記憶装置が、他の記憶装置（ファイルサーバ）が管理するファイルシステム内の領域を利用し、その記憶装置が提供する名前空間の記憶容量を増加させる手段を有することにより達成される。

【0015】また、前記目的は、前述において、前記記憶装置が保持するファイルシステム毎に独立にファイル利用量やファイル作成数に関する上限管理を行う手段と、他の記憶装置上のファイル利用量やファイル作成数

に関する上限管理の管理データを更新する手段と、他の記憶装置上のファイル利用量やファイル作成数に関する上限管理の管理データの最適割り当て値を動的に計算する手段とを有し、さらに、前記憶装置が保持するファイルシステムが、他の記憶装置が保持するファイルシステム上のファイルやディレクトリの実体を指し示す名前を作成する手段と、他の記憶装置から指し示される記憶装置が保持するファイルシステム上のファイルやディレクトリの実体を管理する手段とを有することにより達成される。

【0016】

【発明の実施の形態】以下、本発明による情報処理システムの実施形態を図面により詳細に説明する。

【0017】図1は本発明の第1の実施形態によるファイルサーバを含む情報処理システムの構成を示すブロック図である。ここで説明する本発明の第1の実施形態は、容量が不足したファイルサーバが他のファイルサーバ上にファイルを割り付け、それを容量が不足したファイルサーバがローカルに保持する外部記憶装置と等価に利用することにより、ファイルサーバの拡張記憶領域を確保するようにしたものである。この本発明の第1の実施形態において、1つのファイルシステムによりファイルの記憶管理を実行するため、ファイルシステムが提供する管理機能をそのまま利用することができる。

【0018】図1に示すシステムは、第1のファイルサーバ10a、第2のファイルサーバ10b、第1のファイル共有クライアント12c、第2のファイル共有クライアント12dを備えて構成されている。なお、図1に示す例は、ファイルサーバを2台、ファイル共有クライアントを2台備えるとして示しているが、さらに多くのファイルサーバとファイル共有クライアントとを備えてもよい。

【0019】ファイルサーバ10a、10b、ファイル共有クライアント12c、12dは、それぞれ、中央処理装置（CPU）30a、30b、30c、30d、メモリ32a、32b、32c、32d、ネットワークコントローラ34a、34b、34c、34dを備えて構成されている。また、各ファイルサーバと各ファイル共有クライアントは、ネットワーク14を介して相互に接続されており、それぞれのネットワークコントローラ34a、34b、34c、34dを介して通信を行うことができる。また、ファイルサーバ10a、10bは、それぞれディスクコントローラ36a、36bと物理的記憶装置である複数の外部記憶装置39とを有し、ディスクコントローラ36a、36bと外部記憶装置39とはそれぞれ外部記憶装置接続媒体38a、38bを介して接続されている。

【0020】ファイルサーバ10a、10bのそれぞれのメモリ32a、32b上には、オペレーティングシステム（OS）40a、40bが実行可能に格納されてい

る。オペレーティングシステム40a、40bは、さらに、ファイルシステム管理部42、ファイル共有関係要求解釈部44、ファイル共有外部アクセス実行部46、論理デバイス管理部48を有している。

【0021】図示実施形態は、オペレーティングシステム40a、40bの中でファイルサーバを実現するために必要な機能であるファイルシステム管理部42、ファイル共有関係要求解釈部44、ファイル共有外部アクセス実行部46、論理デバイス管理部48を有しているとしたが、これらの機能の一部または全部は、オペレーティングシステム40a、40bの外に設けられていてもよい。

【0022】ファイル共有クライアント12c、12dのそれぞれのメモリ32c、32d上には、オペレーティングシステム（OS）41c、41dとアプリケーションプログラム60c、60dとがそれぞれ実行可能に格納されている。また、オペレーティングシステム41c、41dは、ファイル共有外部アクセス実行部46を有している。

【0023】ファイルサーバ10a、10b、ファイル共有クライアント12c、12d上のファイル共有外部アクセス実行部46は、ファイルサーバ10a、10bの中のファイル共有関係要求解釈部44で解釈可能なファイル共有に関する処理要求を作成する部分である。ここで作成されたファイル共有処理要求は、ネットワークコントローラ34a、34b、34c、34d、ネットワーク14を介してファイル共有処理を要求するファイルサーバ10a、10bのどちらか適切な方に転送される。ファイル共有外部アクセス実行部46は、その結果を受け取り、結果毎に必要な後処理を実行し、ある一定期間返答がない場合、通信エラーが発生したものと取り扱う。

【0024】ファイルサーバ10a、10b内に存在するファイル共有関係要求解釈部44は、ファイル共有クライアント12c、12dからネットワーク14を通じてネットワークコントローラ34a、34bに到着したファイル共有に関する処理要求を解釈し、必要に応じてファイルシステム管理部42に対して要求を適当な形に変換して転送し、その結果を適当な形に変換して処理要求元に返却する。

【0025】ファイルサーバ10a、10b内のファイルシステム管理部42は、論理デバイス管理部48により提供される論理デバイス52a、52b上にファイルサーバが外部に提供する名前空間やファイルの実体を管理するファイルシステム54a、54bをそれぞれ形成し、ファイルシステム54a、54b上に記録されたファイル56a、56bに対するアクセス実行を行う機能を有し、1つの論理デバイスに対して1つのファイルシステムを割り当てる。ファイル56a、56bに対するアクセス要求がファイルシステム管理部42に到着した

場合、ファイルシステム管理部 42 は、論理デバイス管理部 48 に対して論理デバイス 52 a、52 b 上に存在する必要なデータの取得を要求し、その取得データを元にアクセス要求処理を実行する。

【0026】説明している実施形態におけるファイルシステム管理部 42 は、特定の所有者や特定のグループにより所有されるファイルの利用量や作成ファイル数に上限をかける機能を有するものとする。また、ファイルシステム管理部 42 は、あるファイル実体に対し、完全に等価に扱うことができる複数の名前を付加する機能を有するものとする。さらに、ファイルシステム管理部 42 は、論理デバイス 52 a、52 b の領域を拡張した場合に、動的にファイルシステム 54 a、54 b を拡張し、また、論理デバイス 52 a、52 b の論理ブロック番号的に最後尾からある一定量連続した領域においてファイルシステム 54 a、54 b 中のファイルやディレクトリに関するデータが記録されていない場合に、そのデータが記録されていない部分を動的に縮小する機能を有するものとする。

【0027】論理デバイス管理部 48 は、複数の外部記憶装置 39 上の記憶領域、あるいは、ファイルサーバ 10 a におけるファイルサーバ 10 b 上に記憶されているファイル 56 b のような他のファイルサーバに記憶されたファイルをまとめて論理デバイス 52 a、52 b を形成する。図 1 では、論理デバイス 52 a がこれに相当するものとする。論理デバイスの領域がどの外部記憶装置のどの位置に記憶されるかを管理するため、論理デバイス管理部 48 は、マッピング管理情報 50 を保持する。この情報は、失われると処理を継続できなくなるため、外部記憶装置 39 の特定の領域等の不揮発なメモリ領域にコピーを生成しておく。

【0028】ファイルシステム管理部 42 で作成された論理デバイス 52 a、52 b に対するアクセス要求は、論理デバイス管理部 48 に送られる。論理デバイス管理部 48 は、マッピング管理情報 50 を参照してアクセス要求が行われた領域が実際に記憶されている外部記憶装置または他ファイルサーバ上のファイル、そして、そのデータが記憶されている領域を把握し、そに対してアクセスを実行する。論理デバイス管理部 48 は、外部記憶装置 39 上に必要とされるデータが記憶されている場合、ディスクコントローラ 36 a、36 b を用いて外部記憶装置 39 にアクセスを行う。

【0029】論理デバイス 52 a 上の必要とされるデータがファイル 56 b に存在する場合、そのデータの取得のために、ファイル共有外部アクセス実行部 46 にファイル 56 b 上の必要な領域へのアクセスを依頼する。このとき、外部記憶装置 39 へのアクセス単位はブロック、ファイル 56 b に対するアクセス単位はバイト等のインターフェイスに若干違いがある場合があるが、その差は論理デバイス管理部 48 で吸収する。

【0030】図 2 はマッピング管理情報 50 について説明する図である。図 2 に示す例は、1 つのファイルサーバ上に 2 つの論理デバイスが作成されている場合を示している。マッピング管理情報の中には、構成されている論理デバイスを識別するための名前である論理デバイス名 501、論理デバイスの領域を指定する論理ブロック番号 502、データの記憶先がファイルサーバローカルに利用している外部記憶装置 39（物理デバイス）か、それとも他のファイルサーバ上のファイル（外部ファイル）かを示す記憶デバイスタイプ 503、そのデータが記憶されている物理デバイス名あるいはサーバとそのファイル名である物理デバイス名／サーバ：ファイル名 504、記憶デバイス／外部ファイル中の記録位置を示す物理ブロック番号あるいはファイル内ブロック番号示す物理ブロック番号／ファイル内ブロック番号 505 の各項目が保持される。説明している実施形態は、論理デバイス、データ記憶先タイプ、記憶デバイスの識別に名前を用いているが、その代わりにオペレーティングシステム 40 a、40 b の内部で利用される識別子を用いるようにすることもできる。

【0031】図 1 に示すシステム構成の場合、ファイルサーバは 2 つしか存在しないが、実際にはさらに多数のファイルサーバが存在することも可能である。このとき、マッピング管理情報 50 は、複数のファイルサーバに対して領域拡張用の外部ファイルを割り当てて管理することが可能であり、また、同じファイルサーバに対して複数の領域拡張用の外部ファイルを割り当てて管理することが可能である。

【0032】図 3 はファイルサーバ 10 a でファイル共有関係の処理要求を受け取った後の処理動作を説明するフローチャートであり、以下、これについて説明する。このフローは、ファイルサーバ 10 a 上でファイル共有関係要求解釈部 44 がファイル共有に関するファイル読み出し要求を受け取った後における、ファイルシステム管理部 42、ファイル共有関係要求解釈部 44、ファイル共有外部アクセス実行部 46、論理デバイス管理部 48 の間の処理の流れの概略を示したものである。このフローの例は読み出し処理におけるものであるが、他の処理においてもほぼ同様の処理の流れとなる。

【0033】(1) まず、ファイル共有関係要求解釈部 44 がファイル共有に関するファイル読み出し要求を受け取る。ファイル共有関係要求解釈部 44 は、これにより、ファイルシステム管理部 42 に対して読み出し要求のあったファイルの該当部分の読み出しを依頼する（ステップ 1001、1002）。

【0034】(2) ファイルシステム管理部 42 は、ステップ 1002 での読み出しの依頼を受けると、該当部分がキャッシュされているか否かの確認を行い、キャッシュされている場合、論理デバイスからの読み出しが不要と判断して、後述するステップ 1009 の処理に進

み、キャッシュされていない場合、論理デバイスからの読み出しが必要であるとして、論理デバイス管理部 48 に必要となるデータの読み出しの依頼を行う（ステップ 1003、1004）。

【0035】（3）次に、論理デバイス管理部 48 は、マッピング管理情報 50 を参照して要求されたデータが外部記憶装置 39 上に存在するか、あるいは、他のファイルサーバ上に存在する外部ファイル内に記憶されているかを確認する。この結果、外部記憶装置 39 上にデータが存在した場合、ディスクコントローラ 36a を通して外部記憶装置 39 上のデータを読み出す（ステップ 1005、1006）。

【0036】（4）ステップ 1005 の確認で、要求されたデータが他のファイルサーバ上に存在した場合、論理デバイス管理部 48 は、ファイル共有外部アクセス実行部 46 にたいして、その外部ファイル上の該当データの読み出しを依頼する（ステップ 1007）。

【0037】なお、場合によっては、要求データの記憶先が外部記憶装置 39 上と外部ファイル内とに分割されて記憶されている場合もあり、その場合には、ステップ 1006 とステップ 1007 との双方の処理が必要となる。

【0038】（5）ステップ 1006、1007 の処理で外部記憶装置 39 や外部ファイルからのデータ読み出しが完了後、論理デバイス管理部 48 は、読み出しデータをファイルシステム管理部 42 に返却する（ステップ 1008）。

【0039】（6）ステップ 1008 の処理後、あるいは、ステップ 1003 で、読み出しの依頼による該当部分がキャッシュされていた場合、ファイルシステム管理部 42 は、ファイル共有関係要求解釈部 44 に対して要求されたファイルのデータを返却する（ステップ 1009）。

【0040】（7）次に、ファイル共有関係要求解釈部 44 は、読み出したファイルのデータを処理要求元へ返却し、ここでのファイル共有読み出し処理要求を終了する（ステップ 1010、1011）。

【0041】図 4 は他のファイルサーバ上に存在するファイルを新規作成し、あるいは、既存ファイルを拡張して利用することによりファイルサーバがファイルシステムの領域を拡張する処理動作を説明するフローチャートであり、以下、これについて説明する。図 4 に示すフローは、領域拡張のために領域を確保する確保先ファイルサーバが予め決定されているものとする。この選択方法としては、種々の方法考えられるが、原則的には記憶容量と性能とに余裕があるものが選択される。

【0042】（1）外部ファイル利用による領域拡張処理が開始されると、まず、拡張領域に利用する外部ファイルを新規作成あるいは既に存在している外部ファイルの拡張を行うファイルサーバについて、そのファイルサ

ーバが利用している記憶領域のうち、そのファイルサーバがローカルに利用する外部記憶装置に割り付けられている領域に関して、まだファイルを記憶するために利用されていない空き容量がどの程度存在するか確認する。この確認は、論理デバイス管理部 48 が保持しているマッピング管理情報 50 を取得して、ファイルサーバがローカルに利用している論理ブロック番号を把握し、その領域に関してファイルシステムが管理している領域利用情報を参照して目的の空き容量の残量を取得することにより行われる（ステップ 1101、1102）。

【0043】（2）ステップ 1102 で求めた空き容量が十分であるか否かを判断し、空き容量が拡張のために新たに割り付ける容量に比較して十分な余裕がない場合、領域拡張処理が失敗したとして、ここでの処理を終了する（ステップ 1103、1113）。

【0044】（3）ステップ 1103 での判断で、空き容量が十分存在した場合、領域拡張を行うファイルサーバに対して領域を確保する場合には、その既存外部ファイルを拡張したい容量の分だけ既存の外部ファイルを拡張し、まだ、外部ファイルを割り付けていないファイルサーバに領域を確保する場合には、拡張したい容量の大きさを持つファイルを新規に作成する。このとき、新たに割り付けた領域に対しては初期化のための書き込みを実行しておく（ステップ 1104）。

【0045】（4）次に、外部ファイルの拡張、あるいは、新規作成処理が成功したか否かを確認し、失敗していた場合、領域拡張処理が失敗したとして処理を終了する（ステップ 1105、1113）。

【0046】（5）ステップ 1105 の確認で、新規作成処理が成功していた場合、外部ファイルの新規に割り当てられた領域がどこに割り当てられたかを確認する。新規に割り当てた領域は、割り付け先ファイルサーバが利用している外部ファイル上に記録される可能性がある。この場合、更に別のファイルサーバへのアクセスを行わないと記憶データにアクセスできないことになり、性能的に問題となる。

【0047】ファイルシステムにおいては、ファイルのデータの記憶先は記憶位置管理情報を保持して管理している。そこで、データ実体を任意の空き領域にコピーし、その後に記憶位置管理情報を更新することにより、ファイルシステム内でデータ記録位置の変更を行うことができる。なお、このデータ移動処理実行中には、データの更新異常が発生しないようにするために、移動中のデータに対する更新処理が行われないように管理する必要がある。

【0048】そこで、外部ファイルの新規に割り当てられた領域がどこに割り当てられたかを確認し、もし、割り付け先ファイルサーバが利用する外部ファイルに対して領域が割り当てられた部分が存在する場合には、それ

を割り付け先ファイルサーバがローカルに利用する外部記憶装置に割り付けられている領域に移動する処理を行う。割り付け先領域の確認は、論理デバイス管理部が保持しているマッピング管理情報を取得して他のファイルサーバ上の外部ファイルとして記憶される論理ブロック番号を把握し、ファイルシステムの記憶位置管理情報を参照し、新規に割り付けられた領域が記憶されるデータのブロック番号がその領域に該当するかどうかの確認を行う(ステップ1106)。

【0049】(6)次に、ステップ1106の処理が正常に終了して、外部ファイルが全て割り付け先ファイルサーバのローカルな領域に記憶されているか否かを確認し、全てが割り付け先ファイルサーバのローカルな領域に記憶されていない場合、新たに割り付けを行った領域の開放を行うUndo処理を行った後、領域拡張処理が失敗したとして処理を終了する(ステップ1107、1112、1113)。

【0050】(7)ステップ1107の確認で、外部ファイルが全てが割り付け先ファイルサーバのローカルな領域に記憶されている場合、ファイルシステム拡張のための領域確保が完了していることになるので、確保した領域に関する情報をマッピング管理情報に反映させる(ステップ1108)。

【0051】(8)次に、拡張された領域に対して、ファイルシステムの機能を利用してファイルシステムを拡張する処理を実行し、実行したファイルシステム拡張処理が成功したか否かを確認する。この確認で、ファイルシステムの拡張処理が失敗していた場合、マッピング管理情報を元に戻し、新たに割り付けを行った領域の開放を行うUndo処理を行った後、領域拡張処理が失敗したとして処理を終了する(ステップ1109、1110、1112、1113)。

【0052】(9)ステップ1110の確認で、実行したファイルシステム拡張処理が成功していた場合、領域拡張処理が成功したとして処理を終了する(ステップ1111)。

【0053】図5は他のファイルサーバ上に存在するファイルサーバの領域拡張用の外部ファイルの大きさを縮小する処理動作を説明するフローチャートであり、以下、これについて説明する。ここでの処理において、縮小を行う領域量は予め決定されているものとする。この領域量は、ファイルシステムによりその縮小処理を行うことができる量である必要がある。また、縮小が行われる外部ファイルは、領域縮小を行うファイルシステムを保持する論理デバイスの論理ブロック番号的に最後尾から連続したある縮小を行う領域量以上の領域を保持しているものとする。

【0054】(1)外部ファイルの縮小(外部ファイルの領域の一部開放)処理が開始されると、まず、領域を縮小するファイルシステムの空き容量を確認する。この

確認は、ファイルシステムの空き領域を管理する情報を参照することにより行われる(ステップ1201、1202)。

【0055】(2)ステップ1202の確認で求めた空き容量が開放しようとする領域量以上存在するか否かを確認する。空き容量が開放領域量以上ない場合、そのまま領域開放処理を行うとファイルシステム上の一部のデータが失われることになるため、領域開放処理が失敗したとして処理を終了する(ステップ1203、1211)。

【0056】(3)ステップ1202の確認で求めた空き容量が開放領域量以上存在していた場合、開放される領域中に記録されているファイルやディレクトリのデータが存在していれば、それらを他の領域へ移動する処理を行う。データの存在の有無は、ファイルシステムの記憶位置管理情報を参照することにより開放領域にデータが存在するか否かの確認により行い、データの記録位置変更処理は、図4により説明したステップ1106の処理と同様の処理により行うことができる(ステップ1204)。

【0057】(4)ステップ1204の処理で、全てのデータが開放領域から他の領域に移動されたか否かの確認を行う。この確認で、一部のデータが開放する領域に残っていた場合、処理の続行により一部データが失われることになるため、領域開放処理が失敗したとして処理を終了する(ステップ1205、1211)。

【0058】(5)ステップ1205の確認で、全てのデータが開放領域から他の領域に移動されていた場合、ファイルシステムの機能を利用してファイルシステムの縮小処理を行う。そして、このファイルシステムの縮小処理が成功したか否かを確認する。処理が失敗していた場合、領域開放処理が失敗したとしてここでの処理を終了する(ステップ1206、1207、1211)。

【0059】(6)ステップ1207の確認で、処理が成功していた場合、開放される領域が利用されることはないため、まず、マッピング管理情報から、開放される領域の部分に対するマッピング情報を削除するマッピング管理情報更新処理を行い、ファイルサーバ上に割り付けられている外部ファイルのうち、今回領域が開放された部分について、ファイルのサイズ縮小、あるいは、ファイルの削除によりその領域を開放して、領域の一部開放処理が成功したとして処理を終了する(ステップ1208~1210)。

【0060】図1により説明したシステムの構成は、ファイルサーバを2つ存在させているが、実際には、さらに多数のファイルサーバを設け、自分自身以外の他の任意のファイルサーバに領域拡張用の外部ファイルを割り当てることができる。このとき、ファイルサーバが他のファイルサーバ上に割り当てている領域拡張用の外部ファイルをさらに第3のファイルサーバに移動させること

も可能である。

【0061】図6はファイルサーバが他のファイルサーバ上に割り当てている領域拡張用の外部ファイルをさらに第3のファイルサーバに移動させる処理動作を説明するフローチャートであり、次に、これについて説明する。

【0062】(1) 外部ファイルのサーバ間移動処理が開始されると、まず、ファイルを移動する移動先ファイルサーバにおいて、そのファイルサーバがローカルに利用する外部記憶装置に割り付けられている領域に関して、まだファイルを記憶するために利用されていない空き容量がどの程度存在するかを確認する。この処理は、図4により説明したステップ1102の処理による方法を用いて実行する(ステップ1301、1302)。

【0063】(2) 次に、ステップ1302で求めた空き容量が十分であるか否かを判断し、空き容量が拡張のために新たに割り付ける容量に比較して十分な余裕がない場合、ファイルの移動処理が失敗したとして処理を終了する(ステップ1303、1315)。

【0064】(3) ステップ1303の判断で、空き容量が十分存在する場合、移動するファイルに対して更新がかからないようにするための更新ロックを設定する。これは、移動のためのコピー処理実行中にファイルに更新がかかると、最新のデータの反映処理が複雑になるためである(ステップ1304)。

【0065】(4) 次に、移動を行うファイルをコピー先にコピーする処理を行し、このコピー処理が成功したか否かの確認を行う。この確認で、コピー処理が成功していた場合、移動したファイルがどの領域に割り当てられたか確認する。もし移動先ファイルサーバが利用する外部ファイルに対して領域が割り当てられた部分が存在する場合、それを移動先ファイルサーバがローカルに利用する外部記憶装置に割り付けられている領域に移動する処理を行う。この処理は、図4により説明したステップ1106の処理による方法を用いて実行する(ステップ1305~1307)。

【0066】(5) 次に、ステップ1307での処理が正常に終了して移動した外部ファイルが全て移動先ファイルサーバのローカルな領域に記憶されているか否かを確認する。この確認で、全てが移動先ファイルサーバローカルに記憶されていた場合、移動を行っていたファイルを移動先ファイルサーバのそのファイルサーバがローカルに管理している外部記憶装置に割り当て得られる領域にコピーする処理が完了していることになるので、マッピング管理情報を修正し、移動先を今後参照することが可能にする(ステップ1308、1309)。

【0067】(6) その後、ステップ1304において取得した更新ロックを解除すると共に、移動元ファイルを削除してその領域を開放し、ファイル移動処理が成功したとして処理を終了する(ステップ1310~131

2)。

【0068】(7) ステップ1306でのコピー処理が成功したか否かの確認で、コピー処理が失敗していた場合、また、ステップ1308の確認で移動した外部ファイルが全て移動先ファイルサーバのローカルな領域に記憶されていなかった場合、ステップ1304の処理で取得した更新ロックを解除し、コピー先にコピーされたデータが残っている場合にそれを削除して、ファイルの移動処理が失敗したとして処理を終了する(ステップ1313~1315)。

【0069】前述したファイルサーバが他のファイルサーバ上に割り当てている領域拡張用の外部ファイルを更に第3のファイルサーバに移動させる処理は、データの更新ロックを取得するため、処理実行中に通常処理への影響が大きい。移動処理時間は、移動するファイルの大きさに依存し、大きいほど所要時間が長い。そのため、高頻度で領域拡張用の外部ファイルの移動が行われると考えられる場合、あまりサイズが大きいファイルを多数利用するようにするとよい。このようにすることにより、移動処理を分割して実行することができるようになり、また、1回の移動により影響を受ける範囲を小さくすることができる。さらに、小さな単位で移動ができるようになるので、領域利用の自由度を大きくすることができる。

【0070】ファイルサーバが他のファイルサーバ上に割り当てている領域拡張用の外部ファイルに記憶されているデータをアクセスするには、他のファイルサーバへのアクセスを必要とする。そのため、このようなデータへのアクセスは、ファイルサーバがローカルに管理している外部記憶装置上に記憶されているデータにアクセスする場合に比べて実行時間が長くなる可能性が高い。

【0071】一般に、ファイルシステムは、最終アクセス時刻を保持しているものが多い。この最終アクセス時刻が現時刻に近いものほどアクセス頻度が高い可能性が大きい。そこで、ファイルの最終アクセス時刻を確認して、それが現時刻に近いファイルのデータをファイルサーバがローカルに管理している外部記憶装置上に記憶される領域に記憶し、現時刻から遠いファイルのデータをファイルサーバが他のファイルサーバ上に割り当てている領域拡張用の外部ファイルに記憶されるようにデータの移動処理を適宜実施するとよい。これにより、ファイルサーバの性能を向上させることができる。この場合の移動処理は、図4のステップ1106で説明した方法を用いて実行することができる。

【0072】次に、本発明の第2の実施形態について説明する。本発明の第2の実施形態は、名前空間中のファイル名やディレクトリ名に対して他ファイルサーバへの外部リンク機能を提供し、それにより、ファイル実体やディレクトリを複数のファイルサーバに分散化させることを可能にし、あるファイルサーバにより提供される名

前空間によるファイル記憶容量を増加させるようにしたものである。また、第2の実施形態は、この外部リンクを適切に管理することにより、同一ファイル実体への等価に扱うことができる異なる複数の名前を付加する機能を複数ファイルサーバに跨って提供する。そして、ユーザやグループに対するファイル利用量やファイル作成数の管理は、各ファイルサーバに割り当てを決定し、それぞれファイルサーバに独立に管理させる。また、特定のファイルサーバ内において制限値一杯になった場合には、割り当て量の再割り当てを実施するようにしている。

【0073】図7は本発明の第2の実施形態によるファイルサーバを含む情報処理システムの構成を示すブロック図である。

【0074】図7に示すシステムは、第1のファイルサーバ110a、第2のファイルサーバ110b、第1のファイル共有クライアント112c、第2のファイル共有クライアント112dを備えて構成されている。なお、図7に示す例は、ファイルサーバを2台、ファイル共有クライアントを2台備えるとして示しているが、さらに多くのファイルサーバとファイル共有クライアントとを備えてもよい。

【0075】ファイルサーバ110a、110b、ファイル共有クライアント112c、112dは、それぞれ中央処理装置(CPU)130a、130b、130c、130d、メモリ132a、132b、132c、132d、ネットワークコントローラ134a、134b、134c、134dを備えて構成されている。また、各ファイルサーバと各ファイル共有クライアントは、ネットワーク114を介して相互に接続されており、それぞれのネットワークコントローラ134a、134b、134c、134dを介して通信を行うことができる。また、ファイルサーバ110a、110bは、それぞれディスクコントローラ136a、136bと複数の外部記憶装置139とを有し、ディスクコントローラ136a、136bと外部記憶装置139とはそれぞれ外部記憶装置接続媒体138a、138bを介して接続されている。

【0076】ファイルサーバ110a、110bのそれぞれのメモリ132a、132b上には、オペレーティングシステム(OS)140a、140bが実行可能に格納されている。オペレーティングシステム140a、140bは、さらに、ファイルシステム管理部142、ファイル共有関係要求解釈部144、論理デバイス管理部148を有している。

【0077】図示実施形態は、オペレーティングシステム140a、140bの中でファイルサーバを実現するために必要な機能であるファイルシステム管理部142、ファイル共有関係要求解釈部144、論理デバイス管理部148を有しているとしたが、これらの機能の一

部または全部は、オペレーティングシステム140a、140bの外に有する構成も考えられる。

【0078】ファイル共有クライアント112c、112dのそれぞれのメモリ132c、132d上には、オペレーティングシステム(OS)141c、141dとアプリケーションプログラム170c、170dとがそれぞれ実行可能に格納されている。また、オペレーティングシステム141c、141dは、ファイル共有外部アクセス実行部146を有している。

10 【0079】ファイル共有クライアント112c、112d上のファイル共有外部アクセス実行部146は、ファイルサーバ110a、110b中のファイル共有関係要求解釈部144で解釈可能なファイル共有に関する処理要求を作成する部分である。ここで作成されたファイル共有処理要求は、ネットワークコントローラ134a、134b、ネットワーク114を介してファイル共有処理を要求するファイルサーバ110a、110bのどちらか適切な方に転送される。ファイル共有外部アクセス実行部146は、その結果を受け取り、結果毎に必要となる後処理を実行し、また、ある一定期間返答が内
20 場合、通信エラーが発生したものとして取り扱う。

【0080】ファイルサーバ110a、110b内に存在するファイル共有関係要求解釈部144は、ファイル共有クライアント112c、112d、あるいは、他のファイルサーバ110a、110bからネットワーク114を通じてネットワークコントローラ134a、134bに到着したファイル共有に関する処理要求を解釈し、必要に応じてファイルシステム管理部142に対して要求を適当な形に変換して転送し、その結果を適当な形に変換して処理要求元に返却する。また、ファイル共有関係要求解釈部144は、ファイルサーバ内で処理が閉じない場合、ネットワークコントローラ134a、134b、ネットワーク114を通じて他ファイルサーバに処理要求を転送する。
30

【0081】ファイルサーバ110a、110b内のファイルシステム管理部142は、論理デバイス管理部148により提供される論理デバイス152a、152b上にファイルサーバがディレクトリやファイルの管理を行うファイルシステム154a、154bをそれぞれ形成する。ファイルシステム154a、154b内は、それぞれ独立してファイル利用量その他の管理を行うディレクトリツリー156a、156b、156c、156dに分割される。ファイルシステム管理部142は、ディレクトリツリー156a、156b、156c、156d内に記録されたファイル162a、162b、162c、162dに対するアクセス実行を行う機能を有する。ファイルシステムは、1つの論理デバイスに対して1つのファイルシステムを割り当てる。ファイル162a、162b、162c、162dに対するアクセス要求がファイルシステム管理部142に到着した場合、フ
50

ファイルシステム管理部 142 は、論理デバイス管理部 148 に対して論理デバイス 152 a、152 b 上に存在する必要なデータの取得を要求し、その取得データを元にアクセス要求処理を実行する。

【0082】ファイルサーバ 110 a、110 b 内に存在するファイルシステム管理部 142 は、ファイルシステム管理のために必要な場合、ネットワークコントローラ 134 a、134 b、ネットワーク 114 を通じて他ファイルサーバに処理要求を転送する。

【0083】ネットワーク 114、ネットワークコントローラ 134 a、134 b を通じて他ファイルサーバから送られてきたファイルファイルシステム管理のために必要な処理要求は、ファイルシステム管理部 142 によって受け取られ、必要な処理が実行されて、その結果が返却される。

【0084】本発明の第 2 の実施形態におけるファイルシステム管理部 142 は、ディレクトリ中の名前エントリの中に、自分自身を示すエントリと親ディレクトリを示すエントリとを特定の名称で保持する。このとき、ファイルシステム全体のルートディレクトリの親ディレク

トリは自分自身とする。

【0085】また、ファイルシステム管理部 142 は、自ファイルシステム内において、あるファイル実体に対し完全に等価に扱うことができる複数の名前を付加する機能を有するものとする。但し、ディレクトリに対して、自分自身や親ディレクトリであることを示す特定の名前を除き、複数の名前を作成した場合、ファイルシステムの整合性が保持されないものとする。そして、ファイル実体を管理する情報内には、自分に幾つの名前がつけられているかを管理する情報を保持しているものとする。また、あるファイル実体が同一ファイルシステム上に存在するが、異なる名前空間を形成する複数のディレクトリツリーから共有されることは、管理の面で複雑になるため、ファイル実体に対し完全に等価に扱うことができる複数の名前の付加は、同一のディレクトリツリー内からに限定する。この制約により、ファイル実体が所属するディレクトリツリーは高々 1 つとなる。

【0086】本発明の第 2 の実施形態におけるファイルシステム管理部 142 は、ファイルシステム内で分割されたディレクトリツリー 156 a、156 b、156 c、156 d 毎に独立に、ディレクトリツリー全体、あるいは、ディレクトリツリー内の特定の所有者や特定のグループにより所有されるファイルの利用量や作成ファイル数に上限をかける機能を有している。この機能の実装方法については後述する。

【0087】論理デバイス管理部 148 は、物理的記憶装置である複数の外部記憶装置 139 上の領域をまとめて論理デバイス 152 a、152 b を形成する。ファイルシステム管理部 142 で作成された論理デバイス 152 a、152 b に対するアクセス要求は論理デバイス管

理部 148 に送られる。論理デバイス管理部 148 は、マッピング管理情報 150 を参照してアクセス要求が行われた領域が、実際に記憶されている外部記憶装置 139 とそのデータが記憶されている領域とを把握し、ディスクコントローラ 136 a、136 b を用いて外部記憶装置 139 にアクセスを行う。

【0088】ファイルシステム 152 a、152 b 内には、それぞれ、ディレクトリツリー 156 a、156 b、156 c、156 d に対する、ファイルシステム内のファイル利用量やファイル作成数の上限値の割り当てて管理するディレクトリツリー利用量管理情報 158 a、158 b が記憶されている。

【0089】ディレクトリツリー 156 a、156 c 内には、それぞれ、主ツリー管理情報 160 a、160 b が記憶される。主ツリー管理情報 160 a、160 b は、ファイルサーバ 110 a、110 b がそれぞれ主体となって管理する名前空間と名前空間のルートディレクトリを含むディレクトリツリーに対して割り当てられたファイルシステム内のディレクトリツリーを管理するための情報の集合である。

【0090】ディレクトリツリー 156 b、156 d 内には、それぞれ、副ツリー管理情報 161 a、161 b が記憶される。副ツリー管理情報 161 a、161 b は、他のファイルサーバが主体となって管理される名前空間に対して利用されているディレクトリツリーを管理するための情報の集合である。

【0091】図 8 はマッピング管理情報 150 について説明する図である。図 8 に示す例は、1 つのファイルサーバ上に 2 つの論理デバイスが作成されている場合を示している。マッピング管理情報の中には、構成されている論理デバイスを識別するための名前である論理デバイス名 511、論理デバイスの領域を指定する論理ブロック番号 512、データが記憶されている外部記憶装置 139 の物理デバイス名 513、記憶装置中の記録位置を示す物理ブロック番号 514 の各項目が保持される。ここで説明している実施形態は、論理デバイス、記憶デバイスの識別に名前を用いているが、その代わりにオペレーティングシステム 140 a、140 b の内部で利用される識別子を用いることもできる。

【0092】図 9 は主ツリー管理情報 160 の構成を説明する図である。主ツリー管理情報 160 は、名前空間ファイルサーバ利用管理情報 401、名前空間ユーザ・グループ利用量管理情報 402、ディレクトリツリー内ユーザ・グループ利用量管理情報 403 から構成される。

【0093】図 10 は名前空間ファイルサーバ利用管理情報 401 の構成を説明する図である。図 10 に示す例は、そのディレクトリツリーが所属する名前空間に対して、2 つのファイルサーバ上のディレクトリツリーが割り当てられている例である。名前空間ファイルサーバ利

用管理情報 401 には、名前空間が利用するディレクトリツリーに対する識別子であるツリー ID 521、そのディレクトリツリーが存在するファイルサーバ名 522、そのファイルサーバ内でのディレクトリツリーのルートへのパス名であるディレクトリツリールートパス 523 の各項目が記憶される。ツリー ID “0” のエントリは、そのディレクトリツリーが属する名前空間に対してそれを主体となって管理するファイルサーバとその名前空間のルートディレクトリとなるディレクトリを含むディレクトリツリーのパス名を保持する。

【0094】図 11 は名前空間ユーザ・グループ利用量管理情報 402 の構成を説明する図である。図 11 の例は、名前空間全体と、2 ユーザと 2 グループとに対してファイル利用量やファイル作成数の制約が存在する場合を示している。名前空間ユーザ・グループ利用量管理情報の中には、名前空間、ユーザあるいはグループに対する制約かどうかを示すタイプ 531、制約を受けるユーザあるいはグループの識別子である ID 532、名前空間、ユーザあるいはグループが利用することを許されたファイル利用量とファイル作成数である許可最大ファイル利用量 533 及び許可最大ファイル作成数 534、現在、その名前空間がデータの記録に利用しているディレクトリツリーに対してすでに利用するために割り当てているファイル利用量とファイル作成数である現割り当てファイル利用量 535 及び現割り当てファイル作成数 536 の各項目が保存される。

【0095】本発明の実施形態は、ファイル利用量やファイル作成数の制約が存在するユーザやグループに対してのみ名前空間ユーザ・グループ利用量管理情報でその制約条件を管理している。しかし、全てのユーザ ID とグループ ID とに対して制約情報に関するエントリを保持して管理することも可能である。

【0096】図 12 はディレクトリツリー内ユーザ・グループ利用量管理情報 403 の構成を説明する図である。図 12 に示す例は、ディレクトリツリー全体と、2 ユーザと 2 グループとに対してファイル利用量やファイル作成数の制約が存在する場合の例である。ディレクトリツリー内ユーザ・グループ利用量管理情報の中には、ユーザ、あるいは、グループに対する制約か否かを示すタイプ 541、制約を受けるユーザあるいはグループの識別子である ID 542、ディレクトリツリー内でユーザあるいはグループが利用することを許されたファイル利用量とファイル作成数である割り当てファイル利用量 543 及び割り当てファイル作成数 544、現在、その名前空間がファイル・ディレクトリの記録に利用しているディレクトリツリーに対してすでに利用されているファイル利用量とファイル作成数である現ファイル利用量 545 及び現ファイル作成数 546 の各項目が保存される。

【0097】本発明の実施形態は、ファイル利用量やフ

ァイル作成数の制約が存在するディレクトリツリー全体とユーザやグループとに対してのみ、ディレクトリツリー内ユーザ・グループ利用量管理情報でその制約条件を管理しているが、全てのユーザ ID とグループ ID、そしてディレクトリツリー全体に対して制約情報に関するエントリを保持して管理することも可能である。

【0098】図 13 は副ツリー管理情報の構成を説明する図である。副ツリー管理情報 161 は、名前空間ファイルサーバ利用管理情報 401、ディレクトリツリー内ユーザ・グループ利用量管理情報 403 から構成される。

【0099】図 14 はディレクトリツリー利用量管理情報 158a、158b の構成を説明する図である。図 14 に示す例は、1 つのファイルシステム上に 2 つのディレクトリツリーが存在する場合の例である。ディレクトリツリー利用量管理情報の中には、ディレクトリツリーのルートへのパス名であるツリーパス名 551、そのディレクトリツリーを利用している名前空間の管理主体のファイルサーバの名前と管理主体ファイルサーバにおけるその名前空間のルートディレクトリを保持するディレクトリツリーのパス名である主サーバ名：ツリーパス名 552、そのディレクトリツリーに対して許可される最大のファイル利用量と最大のファイル作成数である許可最大ファイル利用量 553 及び許可最大ファイル作成数 554、現時点でディレクトリツリーに対して利用するために割り当てられているファイル利用量とファイル作成数である現割り当てファイル利用量 555 及び現割り当てファイル作成数 556 の各項目が保存される。

【0100】前述した名前空間ユーザ・グループ利用量管理情報 402 とディレクトリツリー内ユーザ・グループ利用量管理情報 403、そして、ディレクトリツリー利用量管理情報 158 の関係は以下の通りである。

【0101】名前空間は、複数のファイルサーバに存在するディレクトリツリーに対してデータを記憶する。そして、名前空間ユーザ・グループ利用量管理情報 402 は、名前空間全体におけるファイル利用量とファイル作成数の管理を行う。一方、ディレクトリツリー内ユーザ・グループ利用量管理情報 403 は、ある名前空間の構成要素であるディレクトリツリーにおけるファイル利用量とファイル作成数との管理を行う。そのため、名前空間ユーザ・グループ利用量管理情報 402 とディレクトリツリー内ユーザ・グループ利用量管理情報 403 との全体、ユーザ、グループの各エントリにおいて、割り当てファイル利用量 543 の総和は、現割り当てファイル利用量 535 と等しい値を取り、割り当てファイル作成数 544 の総和は現割り当てファイル作成数 536 と等しい値を取る。すなわち、名前空間に対しては、この制約を守るように各ファイルサーバのディレクトリツリーにおけるディレクトリツリー内ユーザ・グループ利用量管理情報 403 を変更してファイル利用量とファイル作

10

20

30

40

50

成数とを適切に分配する。

【0102】一方のディレクトリツリー利用量管理情報158は、あるファイルサーバにおけるディレクトリツリーの利用量の管理を行う。そのため、ディレクトリツリー内ユーザ・グループ利用量管理情報403の全体エントリとそのディレクトリツリーが存在するファイルサーバ上のディレクトリツリー利用量管理情報158の対応するディレクトリツリーのエントリとに対して、割り当てファイル利用量543は、現割り当てファイル利用量555と等しい値を取り、割り当てファイル作成数544は、現割り当てファイル作成数556と等しい値を取る。

【0103】名前空間に対して、新規ファイルサーバの新規ディレクトリツリーを追加するには以下の処理を行う。

【0104】まず、追加を行うファイルサーバ上に、名前空間に追加されるディレクトリツリーを新規に割り当てる。そして、ディレクトリツリー利用量管理情報158に対して追加されたディレクトリツリーに対するエントリを作成し、ツリーパス名551、許可最大ファイル利用量553、許可最大ファイル作成数554、現割り当てファイル利用量555、現割り当てファイル作成数556に適当な値を設定する。

【0105】次に、追加が行われる名前空間を主体となって管理するファイルサーバの名前とその名前空間のルートディレクトリを含むディレクトリツリーのパス名とを取得し、それを元にディレクトリツリー利用量管理情報158の主サーバ名：ツリーパス名552の対応する場所にその情報を設定する。そして、追加が行われる名前空間を主体となって管理するファイルサーバから名前空間ファイルサーバ利用管理情報401とディレクトリツリー内ユーザ・グループ利用量管理情報403との雛型を受け取り、新たに利用するディレクトリツリー内に記録する。その後、ディレクトリツリー内のユーザ・グループ利用量管理情報403の全てのエントリにおける割り当てファイル利用量543、割り当てファイル作成数544、現ファイル利用量545、現ファイル作成数546の値を“0”に初期化する。

【0106】その後、追加が行われる名前空間を主体となって管理するファイルサーバは、その名前空間を形成するディレクトリツリーを保持するファイルサーバに対して、名前空間を形成するディレクトリが追加された後の名前空間ファイルサーバに利用管理情報401を転送し、受け取ったファイルサーバは、対応するディレクトリツリー内のその情報を新規に受け取ったものに更新する。

【0107】前述により、名前空間に対して新規ファイルサーバの新規ディレクトリツリーを追加する処理が完了する。このとき、新規に追加されたディレクトリツリーに対するファイル利用量やファイル作成数の割り当て

は0であるので、この後、必要に応じてファイル利用量やファイル作成数の割り当てを変更して実際に利用していく。逆に利用していたディレクトリツリーを名前空間から削除するにはこの逆の処理を行えばよい。

【0108】ディレクトリツリー全体、あるいは、ディレクトリツリー内の特定の所有者や特定のグループにより所有されるファイルの利用量や作成ファイル数に上限をかける機能を実現するため、本発明の第2の実施形態は、ファイルやディレクトリの実体を管理する情報に対して、それがどのディレクトリツリーに所属しているかを示す情報を付加している。また、ディレクトリツリーへの所属情報の代わりに、直接ディレクトリツリー内ユーザ・グループ利用量管理情報403へのポイントを保持することによっても同様の効果を得ることができる。

【0109】ファイルやディレクトリに対する新たなデータ記憶領域を割り付ける際には、ファイルやディレクトリの実体管理情報の中の所属情報を用いて、その所属するディレクトリツリーにおけるディレクトリツリー内のユーザ・グループ利用量管理情報403の値を参照し、全体と、そのファイル・ディレクトリの所有者のユーザID、所有グループのグループIDに対して、割り当てファイル利用量543と現ファイル利用量545とを取得し、現ファイル利用量545と新規割り当て量の和が割り当てファイル利用量543の値を超えないことを確認することにより制約が守られることを確認する。制約が守られている場合、現ファイル利用量545の値を現ファイル利用量545と新規割り当て量の和の値に更新する。

【0110】領域利用量の制約にかかる場合としては、ディレクトリツリー全体の割り当て量の制約にかかる場合と、そのファイル・ディレクトリの所有者、所有グループの制約にかかる場合との2種類がある。このうち、ディレクトリツリー全体の割り当て量の制約に関する確認とその回避処理とを優先して実行し、その後にファイル・ディレクトリの所有者、所有グループの制約の確認とその回避処理とを実行する。

【0111】制約回避処理は、成功する場合と失敗する場合とがある。制約回避処理成功後は再度処理を実行する。このとき、別の制約にかかる可能性がある。その場合には、再度制約回避処理を実行する。制約回避処理に失敗した場合、ファイルシステム空きなし、あるいは、利用量制約にかかったとしてエラー終了する。

【0112】図15はディレクトリツリー全体のファイル利用量の割り当て制約にかかり、ファイル利用量が不足した場合の制約回避処理の動作を説明するフローチャートであり、以下、これについて説明する。

【0113】(1) ファイル利用量不足回避処理が開始されると、まず、ディレクトリツリーが存在するファイルシステム上のディレクトリツリー利用量管理情報158を参照し、そのディレクトリツリーにおける許可最大

ファイル利用量 553 と現割り当てファイル利用量 555 との差を取得して、ファイルシステム内でのディレクトリツリーの拡張可能量を取得する。その後、拡張可能量が制約回避に必要な量に比べて大きいかなんかを確認する（ステップ 2001、2002）。

【0114】（2）ステップ 2002 の確認で、拡張可能量が制約回避に必要な量に比べて大きかった場合、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報 401 からそのディレクトリツリーが属する名前空間を主体として管理するファイルサーバと、そのルートディレクトリを含むディレクトリツリーのパス名とを調べる。そして、この上に記録されている名前空間ユーザ・グループ利用量管理情報 402 中の名前空間全体のエントリ内の許可最大ファイル利用量 533 と現割り当てファイル利用量 535 とを取得し、その差から名前空間全体の未割り当てファイル利用量を取得する。その後、この値が制約回避に必要な量より大きいかなんかを確認する（ステップ 2003）。

【0115】（3）ステップ 2003 の確認で、未割り当てファイル利用量が制約回避に必要な量より大きかった場合、名前空間全体で未割り当ての利用量が制約回避に十分な量存在することを意味する。そこで、この未割り当てな利用量を利用して制約を回避する。ステップ 2003 の処理で未割り当てファイル量は取得済みであるので、この値を超えないように新たにディレクトリツリーに割り当てるファイル利用量を決定する。この新規割り当てファイル利用量を名前空間ユーザ・グループ利用量管理情報 402 中の名前空間全体のエントリ内の現割り当てファイル利用量 535 に加算し、その後、制約回避処理中のディレクトリエントリに対するディレクトリツリー内ユーザ・グループ利用量管理情報 403 中のディレクトリツリー全体のエントリ内の割り当てファイル利用量 543 に新規割り当てファイル利用量を加算する。前述の処理で制約回避処理が成功したことになり、ここでの処理を終了する（ステップ 2008、2013）。

【0116】（4）ステップ 2003 の確認で、未割り当てファイル利用量が制約回避に必要な量より小さかった場合、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報 401 からそのディレクトリツリーが属する名前空間に属する他のディレクトリツリーを保持するファイルサーバとそのディレクトリツリーへのパス名を取得する。また、それぞれが保持するディレクトリツリー内ユーザ・グループ利用量管理情報 403 中のディレクトリツリー全体のエントリ内の割り当てファイル利用量 543 と現ファイル利用量 545 とを取得し、その差から他のそれぞれのディレクトリツリーに存在する割り当て済だがまだ利用されていないファイル利用量を取得する。そして、この値の和が制約回避に必要な量より大きいかなんかを確認する。この確認で、

まだ利用されていないファイル利用量の和が制約回避に必要な量に足りない場合、制約回避処理が失敗したとしてここでの処理を終了する。（ステップ 2004、2014）。

【0117】（5）ステップ 2004 の確認で、まだ利用されていないファイル利用量の和が制約回避に必要な量に十分であった場合、名前空間全体には未割り当てな領域が存在しないが、他のディレクトリツリーには利用割り当てが行われたがまだ利用されていない利用量が制約を回避するに十分な量存在することを意味する。そこで、これらの余裕分をディレクトリツリー全体の利用量の制約にかかったディレクトリツリーで利用することとしてこの制約を回避する。ステップ 2004 の処理ですでに他のディレクトリツリーにおける未利用分が判っているため、その値より小さな値で割り当てファイル利用量の開放を要求する。割り当てファイル利用量の開放要求を受け取ったファイルサーバは、対象となるディレクトリツリーにおけるディレクトリツリー内ユーザ・グループ利用量管理情報 403 中のディレクトリツリー全体のエントリ内の割り当てファイル利用量 543 と現ファイル利用量 545 とを取得し、要求量の分を割り当てファイル利用量 543 から減じても問題がないことを確認した後にその値を要求量の分だけ減算する。割り当てファイル利用量の開放の要求は、制約回避に必要な量に適当なマージンを加えた量の分について行われる。1 つのディレクトリツリーのみで必要量がそろわない場合、複数のディレクトリツリーに割り当てファイル利用量の開放を要求する。必要量の割り当てファイル利用量の開放が完了後、ディレクトリツリー内ユーザ・グループ利用量管理情報 403 中のディレクトリツリー全体のエントリ内の割り当てファイル利用量 543 にその開放量を加算する。前述の処理で制約回避処理が成功したことになり、ここでの処理を終了する（ステップ 2009、2013）。

【0118】前述した処理において、ファイルやディレクトリに対するデータ記憶領域が開放された場合、ファイルやディレクトリの実体管理情報中の所属情報を用いて、その所属するディレクトリツリーにおけるディレクトリツリー内ユーザ・グループ利用量管理情報 403 の値を参照し、全体と、そのファイルの所有者のユーザ ID、所有グループのグループ ID とに対して、現ファイル利用量 545 を取得し、その値を取得値から開放量を減算した値に更新する。また、ファイル作成数に関しても、ファイルやディレクトリの新規作成・削除処理が行われた際に同じ処理フローで対処可能である。

【0119】本発明の実施形態は、ファイルやディレクトリの実体を管理する情報に対して、それがどのディレクトリツリーに所属しているかを示す情報を付加したが、直接ディレクトリツリー内ユーザ・グループ利用量管理情報 403 へのポインタを保持するようにしても同

様の処理を実現することができる。

【0120】(6) ステップ2002の確認で、拡張可能量が制約回避に必要な量に比べて小さかった場合、前述で説明したステップ2003の処理の場合と同様に、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報401からそのディレクトリツリーが属する名前空間を主体として管理するファイルサーバと、そのルートディレクトリを含むディレクトリツリーのパス名とを調べ、その上に記録されている名前空間ユーザ・グループ利用量管理情報402の中の名前空間全体のエントリ内の許可最大ファイル利用量533と現割り当てファイル利用量535とを取得し、その差から名前空間全体の未割り当てファイル利用量を取得する。そして、この値が制約回避に必要な量より大きいかなかを確認する(ステップ2005)。

【0121】(7) ステップ2005の確認で、未割り当てファイル利用量が制約回避に必要な量より大きく、名前空間全体に十分な未割り当てファイル利用量がある場合、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報401からそのディレクトリツリーが属する名前空間に属する他のディレクトリツリーを保持するファイルサーバとそのディレクトリツリーへのパス名とを取得する。そして、それぞれのディレクトリツリーに対して、ファイルサーバにおける割り当てファイル利用量をどの程度拡張可能か確認する。それぞれのディレクトリツリーにおける拡張可能量の取得処理は、ステップ2002で説明した方法を用いる。その後、この拡張可能量の総和が制約回避に必要な量より大きいかなかを確認する(ステップ2006)。

【0122】(8) ステップ2006の確認で、拡張可能量の総和が制約回避に必要な量に足りず、拡張不可能であった場合、または、ステップ2005の確認で、未割り当てファイル利用量が制約回避に必要な量に足りなかった場合、ステップ2004の場合と同様の確認を行う。すなわち、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報401からそのディレクトリツリーが属する名前空間に属する他のディレクトリツリーを保持するファイルサーバとそのディレクトリツリーへのパス名とを取得する。それぞれが保持するディレクトリツリー内ユーザ・グループ利用量管理情報403中のディレクトリツリー全体のエントリ内の割り当てファイル利用量543と現ファイル利用量545とを取得し、その差からそれぞれの他のディレクトリツリーに存在する割り当て済だがまだ利用されていないファイル利用量を取得する。そして、この値の和が制約回避に必要な量より大きいかなかを確認する。この確認で、まだ利用されていないファイル利用量の和が制約回避に必要な量に足りない場合、制約回避処理が失敗したとしてここでの処理を終了する(ステップ2007、2014)。

【0123】(9) ステップ2006の確認で、拡張可能量の総和が制約回避に必要な量より大きく、他のファイルサーバでディレクトリツリー拡張可能であった場合、他ディレクトリツリーに対して制約回避必要量に適当なマージンを足した量のファイル利用量を割り当てる。ステップ2006の処理で、拡張可能なディレクトリツリーと拡張可能量とは判っているので、このファイル利用量が割り当てられた領域に対して後述するステップ2011の処理でデータの移動を行う。1つのディレクトリツリーで必要量の割り当てが完了しない場合、複数のディレクトリツリーに対してファイル利用量の割り当てを実施する。ファイル利用量の割り当て処理の手順は、ステップ2008の処理で実施した方法を用いる(ステップ2010)。

【0124】(10) ステップ2010の処理で他ディレクトリツリーを拡張した後、または、ステップ2007での確認で、まだ利用されていないファイル利用量の和が制約回避に必要な十分な量だけ存在した場合、ディレクトリツリー全体のファイル利用量の制約にかかったディレクトリツリー内の一部のデータを他ディレクトリツリー上に移動することにより、その制約を回避する処理を実行する。データの移動処理の実行方法は後述する。他のディレクトリツリーに存在する割り当て済であるがまだ利用されていないファイル利用量が移動可能なデータ量であるので、ステップ2004で説明した方法で移動先となることができるディレクトリツリーとそこに対する移動可能量を取得する。ステップ2010を経由してこの処理を実施する場合、割り当てファイル利用量の追加処理が行われたディレクトリツリーを優先的に移動先として採用する。そうでない場合、移動可能量が大きいところを優先的に移動先として採用する。1つのディレクトリツリーへの移動によっては制約が完全に回避できない場合、複数のディレクトリツリーに対して移動処理を実行する。データの移動量は、制約回避の必要量に適当なマージンを加えた量とする(ステップ2011)。

【0125】(11) 次に、ステップ2011の処理でのデータの移動処理が成功したかなかの確認を行い、移動処理が成功して、ディレクトリツリーに対する制約が回避された場合、制約回避処理成功としてここでの処理を終了し、そうでない場合、制約回避が不可能であり、制約回避処理が失敗したとしてここでの処理を終了する(ステップ2013、2014)。

【0126】図15では、ディレクトリツリー全体のファイル利用量の割り当て制約にかかった場合の回避処理について説明した。ディレクトリツリー全体のファイル作成数の制約にかかった場合の回避処理についても、これと同じ処理フローで対応可能である。このとき、許可最大ファイル利用量544を許可最大ファイル作成数534に、現割り当てファイル利用量535を現割り当て

ファイル作成数 536 に、割り当てファイル利用量 543 を割り当てファイル作成数 544 に、現ファイル利用量 545 を現ファイル作成数 546 に、許可最大ファイル利用量 553 を許可最大ファイル作成数 554 に、現割り当てファイル利用量 555 を現割り当てファイル作成数 556 にそれぞれ読み替える。

【0127】図 16 はディレクトリツリー内のユーザやグループへのファイル利用量の割り当て制約にかかり、ファイル利用量が不足した場合の制約回避処理の動作を説明するフローチャートであり、以下、これについて説明する。前述したように、この処理が行われるときには、ディレクトリツリー内自体にはすでに記録可能領域が存在する。そこで、ディレクトリツリー間で制約にかかったユーザやグループの割り当て量の調整を行う。

【0128】(1) ディレクトリツリー内のユーザやグループへの割り当て量不足の制約解除処理が開始されると、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報 401 からそのディレクトリツリーが属する名前空間を主体として管理するファイルサーバと、そのルートディレクトリを含むディレクトリツリーのパス名とを調べる。その上に記録されている名前空間ユーザ・グループ利用量管理情報 402 中の制約にかかったユーザ ID またはグループ ID のエントリ内の許可最大ファイル利用量 533 と現割り当てファイル利用量 535 とを取得し、その差から名前空間全体の未割り当てファイル利用量を取得する。そして、この値が制約回避に必要な量より大きいかなんかを確認する（ステップ 2101、2102）。

【0129】(2) ステップ 2102 の確認で、未割り当てファイル利用量が制約回避に必要な量に足りない場合、ディレクトリツリー上に記憶されている名前空間ファイルサーバ利用管理情報 401 からそのディレクトリツリーが属する名前空間に属する他のディレクトリツリーを保持するファイルサーバとそのディレクトリツリーへのパス名を取得する。それぞれが保持するディレクトリツリー内ユーザ・グループ利用量管理情報 403 中の制約にかかったユーザ ID またはグループ ID のエントリ内の割り当てファイル利用量 543 と現ファイル利用量 545 とを取得し、その差からそれぞれの他のディレクトリツリーに存在する割り当て済であるがまだ利用されていないファイル利用量を取得する。この値の和が制約回避に必要な量より大きいかなんかを確認する。この確認で、まだ利用されていないファイル利用量の和が制約回避のための必要量に足りない場合、制約回避が不可能であるため、制約回避処理失敗としてここでの処理を終了する（ステップ 2103、2107）。

【0130】(3) ステップ 2102 の確認で、未割り当てファイル利用量が制約回避に必要な量に十分存在する場合、名前空間全体で未割り当てな利用量が存在することを意味するので、この未割り当てな利用量を利用し

て制約を回避する。ステップ 2102 の処理で未割り当てファイル量は取得済みであるので、この値を超えないように新たにディレクトリツリーに割り当てるファイル利用量を決定する。この新規割り当てファイル利用量を名前空間ユーザ・グループ利用量管理情報 402 中の制約にかかったユーザ ID またはグループ ID のエントリ内の現割り当てファイル利用量 535 に加算し、その後、制約回避処理中のディレクトリエントリに対するディレクトリツリー内ユーザ・グループ利用量管理情報 403 中の制約にかかったユーザ ID またはグループ ID のエントリ内の割り当てファイル利用量 543 に新規割り当てファイル利用量を加算する。その後、ディレクトリツリー内のユーザやグループへの割り当て量不足の制約解除処理が成功したとしてここでの処理を終了する（ステップ 2104、2106）。

【0131】(4) ステップ 2103 の確認で、まだ利用されていないファイル利用量の和が制約回避のための必要量を満たしている場合、名前空間全体には未割り当てな領域が存在しないが、他のディレクトリツリーには利用割り当てが行われたがまだ利用されていない利用量が存在していることを意味する。そこで、これらの余裕分をユーザ ID またはグループ ID の利用量の制約にかかったディレクトリツリーで利用することによりこの制約を回避する。ステップ 2103 での処理ですでに他のディレクトリツリーにおける未利用分が判っているの、その値より小さな値で割り当てファイル利用量の開放を要求する。割り当てファイル利用量の開放要求を受け取ったファイルサーバは、対象となるディレクトリツリーにおけるディレクトリツリー内ユーザ・グループ利用量管理情報 403 中の制約にかかったユーザ ID またはグループ ID のエントリ内の割り当てファイル利用量 543 と現ファイル利用量 545 とを取得し、要求量の分を割り当てファイル利用量 543 から減じても問題がないことを確認した後にその値を要求額分だけ減算する。割り当てファイル利用量の開放の要求は、制約回避に必要な量に適当なマージンを加えた量について行う。1 つのディレクトリツリーのみで必要量がそろわない場合、複数のディレクトリツリーに割り当てファイル利用量の開放を要求する。必要量の割り当てファイル利用量の開放が完了後、ディレクトリツリー内ユーザ・グループ利用量管理情報 403 中の制約にかかったユーザ ID またはグループ ID のエントリ内の割り当てファイル利用量 543 にその開放量を加算する。その後、ディレクトリツリー内のユーザやグループへの割り当て量不足の制約解除処理が成功したとして処理を終了する（ステップ 2105、2106）。

【0132】図 16 では、ディレクトリツリー内のユーザやグループへのファイル利用量の割り当て制約にかかった場合の回避処理について説明した。ディレクトリツリー内のユーザやグループへのファイル作成数の制約に

かかった場合の回避処理についても、これと同じ処理フローで対応可能である。このとき、許可最大ファイル利用量 544 を許可最大ファイル作成数 534 に、現割り当てファイル利用量 535 を現割り当てファイル作成数 536 に、割り当てファイル利用量 543 を割り当てファイル作成数 544 に、現ファイル利用量 545 を現ファイル作成数 546 に、許可最大ファイル利用量 553 を許可最大ファイル作成数 554 に、現割り当てファイル利用量 555 を現割り当てファイル作成数 556 にそれぞれ読み替える。

【0133】本発明の実施形態におけるファイルシステム管理部 142 は、外部リンクの機能を有するものとする。通常、多くのファイルシステムは、ディレクトリの各名前エントリに対して、それがディレクトリかファイルか等の実体の種類を示す情報を保持しており、その種類に対して、さらに他ファイルサーバのファイルシステム上に存在するディレクトリあるいはファイルであることを示すものを追加する。

【0134】通常のディレクトリの名前エントリに対しては、その名前が示すファイルあるいはディレクトリの実体を示す識別子が記録される。外部リンクであるディレクトリの名前エントリに対しては、そのディレクトリやファイルの名前が記憶されているファイルサーバとそのディレクトリパス名、そして、その上での実体を識別するために、名前空間ファイルサーバ利用管理情報 401 のツリー ID 521 と記憶先ファイルサーバにおけるファイルの識別子とを記憶する。

【0135】外部リンク先における名前が判らないとアクセスを行うことができない。そこで、後述する特定の利用法の外部リンクを除き、リンク先における名前の作成法をあらかじめ決めておく必要がある。

【0136】図 17 は外部リンク機能の外部リンク先における被参照名の決定方法について説明する図である。この図において、パス名 901 は、ディレクトリがリンク先の場合の名前を示す。そして、パス名 911 は、外部リンク先が存在するディレクトリツリーのルートディレクトリのパス名、パス名 912 は、ディレクトリツリーにおいて外部からリンクされるファイルやディレクトリを管理するディレクトリを示すディレクトリツリーのルートディレクトリからの相対パス名である。識別子 913 は、この名前の実体がファイルかディレクトリであるかを示す識別子である。識別子 914 は、この名前の実体のファイルシステム内での識別子である。識別子 915 は、このファイルを参照する参照元のツリー ID である。この値は、名前空間ファイルサーバ利用管理情報 401 中のものを利用する。識別子 916 は、このディレクトリやファイルを参照する外部リンクを保持するディレクトリの参照元ファイルシステムにおける識別子である。

【0137】パス名 902 は、ファイルがリンク先の場

合の名前を示す。本発明実施形態は、あるファイル実体に対し完全に等価に扱うことができる複数の名前を付加する機能を実現している。そのため、リンク先がファイルの場合、同一のファイル実体に対して同一ディレクトリ中に複数の名前を作成する。すなわち、同一ディレクトリの中から複数の外部リンクが作成される可能性がある。また、リンク数管理のために、参照されるリンク毎にファイルの名前を作成する。そのため、外部リンクにより参照されるファイル名として、パス名 911、パス名 912、識別子 913、識別子 914、識別子 915、識別子 916 だけでは重複する可能性がある。そのため、これらに加えて、重複を避ける識別子 917 を付加して名前を決定する。

【0138】識別子 917 自体は、リンク先ファイルシステム内で複数のリンクを作成させる以外の意味はないため、ファイル実体を参照する際には、任意の値のものを利用することができる。そこで、この識別子 917 としては、基本的に 0 からの通番を利用し、ファイル実体を参照する際に、識別子 917 として 0 を利用する。リンク先における名前エントリ削除時には、識別子 917 が 0 のものを最後に削除するようにする。

【0139】前述したように、本発明の実施形態におけるファイルシステム管理部 142 は、ディレクトリの中の名前エントリの中に、自分自身を示すエントリと、親ディレクトリを示すエントリとを特定の名称で保持する。また、自分自身と親ディレクトリとして参照される場合を除き、ディレクトリ実体への名前が複数存在した場合、ファイルシステムの整合性は保証されない。そのため、本発明の実施形態は、外部リンクで参照されるディレクトリにおいても、親ディレクトリを 1 つ管理するようにしており、これにより、ファイルシステム管理部 142 と同等の機能を複数ファイルサーバのファイルシステムに跨って実現することが可能となる。そこで、外部リンクとして参照されるディレクトリには、その外部リンク元を示す外部リンクのエントリを特定の名称で作成しておく。そして、ファイル共有クライアントから、その親ディレクトリを示す名前に関する処理要求をファイル共有関係要求解釈部 144 が受け取った場合、ファイルシステム内部の親ディレクトリではなく、外部リンク元へのリンク情報を利用して処理を実行する。

【0140】外部リンクにより参照される実体に対しては、ファイルサーバ内のみで処理を行うことができない。この場合、ファイルシステム管理部 142 は、ファイル共有関係要求解釈部 144 から受け取った要求をアクセス先変更として外部リンク先のファイルサーバ名とそのパス名とを返却する。このパス名は、原則として図 17 において説明した方法で作成されている。ファイル共有関係要求解釈部 144 も、これをそのままアクセス要求を出したファイル共有クライアントへ返却する。この応答を受け取ったファイル共有クライアント内のファ

イル共有外部アクセス実行部 146 は、指定されたパス名に対するアクセスを実行する。

【0141】前述したリンク先におけるパス名の作成において、外部リンクとして参照されるディレクトリに対する親ディレクトリが指定された場合の処理は例外となる。この場合、リンク元のファイルサーバとそのディレクトリ実体の識別子としか判らず、その親ディレクトリへのパス名が不明である。

【0142】ファイルシステム管理部 142 には、ディレクトリ実体において、その親ディレクトリを示す名前エントリが存在する。そのため、ファイルシステム内で親ディレクトリを手繰っていくことによりそのファイルシステムのルートディレクトリにたどりつき、名前を確認しながらそのパスを戻ることにより、リンク元のファイルサーバにおけるディレクトリ実体のパス名を取得することができる。このため、ファイルシステム管理部 142 は、ネットワーク 114、ネットワークコントローラ 134a、134b を通して送られてきた他ファイルサーバからディレクトリ実体の識別子を受け取り、それを元にそのディレクトリ実体へのパス名を作成してネットワークコントローラ 134a、134b、ネットワーク 114 を通して返却する機能を有している。

【0143】ファイルシステム管理部 142 は、外部リンクとして参照されるディレクトリに対する親ディレクトリに関する処理を要求された場合、ネットワークコントローラ 134a、134b、ネットワーク 114 を介してリンク元の外部リンクを保持するファイルサーバに対して、そのディレクトリ実体のパス名を作成する処理要求を送信し、その結果を元にファイル共有関係要求解釈部 144 に結果を返却する。

【0144】図 18 はディレクトリ内の外部リンクである名前エントリを削除する場合の処理動作を説明するフローチャートであり、以下、これについて説明する。

【0145】(1) 外部リンクである名前エントリを削除する処理が開始されると、まず、削除される外部リンクの名前エントリを含むディレクトリが所属する名前空間ファイルサーバ利用管理情報 401 におけるツリー ID と、そのディレクトリの実体の、それが記憶されているファイルサーバのファイルシステムにおける識別子と、リンク先の実体が記憶されているファイルサーバと、そのファイルサーバのファイルシステムにおけるリンク先の実体の識別子とを取得する。また、外部リンクの対象がファイルである場合には、削除される名前エントリを含むディレクトリにおいて、指定ファイル実体に対していくつ外部リンクエントリが存在するか調べてその値も取得する(ステップ 2201、2202)。

【0146】(2) リンク先の実体が記録されているファイルシステムにおけるステップ 2202 の処理で取得した情報を元に外部リンク被参照用の名前を作成し、その名前を削除する。名前が指し示す実体がファイルであ

る場合、名前作成時に、重複を避ける識別子 917 が必要となるが、これは、ステップ 2202 の処理で取得した削除される名前エントリを含むディレクトリにおいて指定ファイル実体に対していくつ外部リンクエントリが存在するか調べたその結果から 1 を減じた値を用いる(ステップ 2203)。

【0147】(3) 削除される外部リンクの名前エントリを含むディレクトリにおいて、指定された名前エントリを削除して処理を完了する(ステップ 2204、2205)。

【0148】外部リンクが作成・更新されるパターンとしては、以下に示すようなパターンが考えられる。すなわち、

パターン 1. あるファイル実体に対して等価に扱うことができる名前を新規に作成する(link処理の場合)、ファイル実体が存在するディレクトリツリーと名前を新規に作成するディレクトリが所属するディレクトリツリーが異なるファイルサーバのファイルシステムに存在する場合、

パターン 2. ファイル、あるいは、ディレクトリの名前を変更する(rename処理の場合)、変更前の名前エントリが外部リンクである、あるいは、変更前の名前エントリを含むディレクトリが所属するディレクトリツリーと変更後の名前エントリを含むディレクトリが所属するディレクトリツリーとが異なるファイルサーバのファイルシステム上に存在する場合、

パターン 3. データの記録位置変更処理を実行した場合である。

【0149】図 19 は前述のパターン 1 のあるファイル実体に対して等価に扱うことができる名前を新規に作成する場合に、ファイル実体が存在するディレクトリツリーと名前を新規に作成するディレクトリが所属するディレクトリツリーが異なるファイルサーバのファイルシステムに存在するときの外部エントリ作成処理の動作を説明するフローチャートである。

【0150】(1) 外部エントリ作成の処理が開始されると、リンク元ディレクトリが所属する名前空間ファイルサーバ利用管理情報 401 におけるツリー ID と、そのディレクトリの実体のそれが記憶されているファイルサーバのファイルシステムにおける識別子と、リンク先ファイル実体が記憶されているファイルサーバと、そのファイルシステムにおけるリンク先ファイル実体の識別子とを取得する。また、リンク元ディレクトリにおいて、すでに指定ファイル実体に対していくつ外部リンクエントリが存在するかを調べてその値をも取得する(ステップ 2301、2302)。

【0151】(2) リンク先ファイルシステムが、指定されたファイル実体に対してステップ 2302 で取得した情報を元に外部リンク被参照用の名前を付加する。この処理は、ファイルサーバが保持するファイルシステム

管理部 142 のあるファイル実体に対し完全に等価に扱うことができる複数の名前を付加する機能を用いて実現される。名前作成時に、重複を避ける識別子 917 が必要となるが、これは、ステップ 2202 で取得したリンク元ディレクトリエントリにおいてすでに指定ファイル実体に対していくつ外部リンクエントリが存在するか調べたその結果の値を用いる（ステップ 2303）。

【0152】（3）次に、リンク元ディレクトリにおいて、ステップ 2302 で取得した情報を用いて指定された名前の外部リンクエントリを作成し、ここでの処理を完了する（ステップ 2304、2305）。

【0153】図 20 はパターン 2 のファイルあるいはディレクトリの名前を変更する（rename 処理）場合に、変更前名前のエントリが外部リンクである、あるいは、変更前の名前エントリを含むディレクトリが所属するディレクトリツリーと変更後の名前エントリを含むディレクトリが所属するディレクトリツリーが異なるファイルサーバのファイルシステム上に存在する場合の名前変更処理の動作を説明するフローチャートであり、以下、これについて説明する。

【0154】ファイル、あるいは、ディレクトリの名前を変更する処理においては、変更前名前が外部リンクではなく、かつ、変更前の名前エントリを含むディレクトリが所属するディレクトリツリーと変更後の名前エントリを含むディレクトリが所属するディレクトリツリーとが同一の場合も存在するが、この場合にはディレクトリツリーが所属するファイルシステム内で名前の変更処理を行うため、外部リンクに関する処理は行われないので図 20 では省略した。

【0155】なお、ここで説明する処理は、名前の変更のみを実行し、ファイルやディレクトリの実体は移動しないものとする。

【0156】（1）名前変更処理が開始されると、変更前の名前のエントリを含むディレクトリが所属する名前空間ファイルサーバ利用管理情報 401 におけるツリー ID と、そのディレクトリの実体のそれが記憶されているファイルサーバのファイルシステムにおける識別子と、変更後に名前のエントリが記憶されるディレクトリが所属する名前空間ファイルサーバ利用管理情報 401 におけるツリー ID と、そのディレクトリの実体のそれが記憶されているファイルサーバのファイルシステムにおける識別子と、名前により示される実体が保存されているディレクトリツリーの名前空間ファイルサーバ利用管理情報 401 におけるツリー ID と、そのファイルシステム中の実体の識別子とを取得する。また、名前変更対象がファイルの場合、変更前の名前エントリを含むディレクトリにおいて、指定ファイル実体に対して外部リンクエントリが幾つ存在するか、また、変更後に名前エントリが記憶されるディレクトリにおいて、すでに指定ファイル実体に対していくつ外部リンクエントリが存在

するか調べてそれらの値も取得する（ステップ 2401、2402）。

【0157】（2）変更前の名前エントリが外部リンクかそうでないかを確認する。この確認で、変更前の名前エントリが外部リンクではない場合、名前変更後は外部リンクとして参照されるような場合を意味する。そのため、現在外部リンクでないものを他ファイルサーバからリンクされる外部リンク化処理を行う。そして、名前変更後に外部リンクの参照に用いられる名前を、ステップ 2402 で取得した情報を元に作成し、実体を記憶するファイルシステムにおいて、変更前の名前からその作成した名前に変更する処理を行う。名前が変更される実体がファイルの場合、変更後の名前である外部リンク被参照用の名前を作成するとき、同じ名前の重複を避けるための識別子 917 が必要となるが、その値には、ステップ 2402 で取得した変更後に名前エントリが記憶されるディレクトリにおいて、すでに指定ファイル実体に対していくつ外部リンクエントリが存在するかを調べた値を識別子 917 として用いる。名前が変更される実体がディレクトリの場合、特定の名称で親ディレクトリを示す外部リンクのエントリを実体内に新規作成する（ステップ 2403、2405）。

【0158】（3）ステップ 2405 の処理終了後、変更後名前を含むディレクトリにおいて、ステップ 2402 で取得した情報を用いて変更後指定名で実体に対する外部リンクのエントリを作成して、名前変更の処理を完了する（ステップ 2406、2412）。

【0159】（3）ステップ 2403 の確認で、変更前の名前エントリが外部リンクであった場合、外部リンクにより指し示される実体が記憶されているディレクトリツリーのツリー ID と、名前変更後に名前エントリが記憶されるディレクトリが所属するディレクトリツリーのツリー ID とを比較して、実体と名前変更先とが同一のディレクトリツリーであるか否かを調べる（ステップ 2404）。

【0160】（4）ステップ 2404 のチェックで、実体と名前変更先とが同一のディレクトリツリーではなかった場合、名前変更前も名前変更後も外部リンクである場合であることを意味する。この処理では、名前変更前に外部リンク参照のために用いる名前と名前変更後に外部リンク参照のために用いる名前とを、ステップ 2402 の処理で取得した情報を元に作成し、実体を記憶するファイルシステムにおいて、変更前のリンクに用いる名前から変更後のリンクに用いる名前に変更する処理を行う。名前が変更される実体がファイルの場合、外部リンク被参照用の名前を作成するとき、同じ名前の重複を避けるための識別子 917 が必要となる。変更前の名前には、ステップ 2402 で取得した変更前に名前エントリが含まれていたディレクトリにおいて、指定ファイル実体に対していくつ外部リンクエントリが存在するかを調

べた値から1を減じた値を、変更後の名前には、ステップ2402で取得した変更後に名前エントリが記憶されるディレクトリにおいて、すでに指定ファイル実体に対していくつ外部リンクエントリが存在するかを調べた値と同じ名前の重複を避けるための識別子917として用いる。名前が変更される実体がディレクトリの場合、特定の名前で実体内に存在する親ディレクトリを示す外部リンクエントリを修正する(ステップ2407)。

【0161】(5) ステップ2407の処理後、変更後名前を含むディレクトリにおいて、ステップ2402で取得した情報を用いて変更後指定名で実体に対する外部リンクを作成し、その後、変更前の名前エントリを含むディレクトリにおいて変更前名前エントリを削除して、名前変更の処理を完了する。なお、前述の削除処理は、ディレクトリ内の名前エントリのみを削除する処理である(ステップ2408、2409、2412)。

【0162】(6) ステップ2404のチェックで、実体と名前変更先とが同一のディレクトリツリーであった場合、名前変更前は外部リンクであるが、名前変更後は外部リンクではなくなる場合であることを意味する。ここでの処理では、名前変更前に外部リンク参照のために用いる名前を、ステップ2402で取得した情報を元に作成し、実体を記憶するファイルシステムにおいて、変更前のリンクに用いる名前から変更後の指定された名前に変更する処理を行う。名前が変更される実体がファイルの場合、変更前の名前である外部リンク被参照用の名前を作成するとき、同じ名前の重複を避けるための識別子917が必要となるが、ステップ2402で取得した変更前に名前エントリが含まれるディレクトリにおいて、指定ファイル実体に対していくつ外部リンクエントリが存在するかを調べた値から1を減じた値を識別子917として用いる。名前が変更される実体がディレクトリの場合、特定の名前で親ディレクトリを示す外部リンクエントリが実体内に存在するが、それを削除する(ステップ2410)。

【0163】(7) その後、変更前の名前エントリを含むディレクトリにおいて変更前名前エントリを削除して、名前変更の処理を完了する。なお、前述の削除処理は、ディレクトリ内の名前エントリのみを削除する処理である(ステップ2411、2412)。

【0164】パターン3のデータの記録位置変更処理を実行した場合におけるデータの記録位置変更処理としては各種のものが考えられる。ここでは、基本的な処理のケースと考えられる2つのケースを考える。1つ目は、あるディレクトリの実体を他のファイルサーバに存在する他のディレクトリツリーに移動することである。2つ目は、あるファイルの実体を他のファイルサーバに存在する他のディレクトリツリーに移動することである。全ての移動は、この2つの移動の組み合わせにより実現することができる。

【0165】図21はディレクトリ実体を別のファイルサーバの上に存在する他のディレクトリツリーへ移動する処理を説明するフローチャートであり、以下、これについて説明する。ここでの処理において、実体を移動するディレクトリは、名前空間全体によるパス名で指定されるものとする。また、移動先は、ファイルサーバとそのファイルサーバ内のディレクトリツリーが指定されるものとする。

【0166】(1) ディレクトリ実体移動処理が開始されると、実体の記憶場所を移動するように指示されたディレクトリに対して、通常アクセスに対するロックを設定する。このロックは、移動処理中に中身の変更が行われないようにするためである。そして、このロックは、実体を管理するデータの中にロックを管理するフラグを設定し、このフラグがセットされている間、アクセスを行わないようにすることにより実現する。但し、このロックが設定されていても、移動処理に関するアクセスは可能である(ステップ2501、2502)。

【0167】(2) 移動先に指定されたファイルサーバのディレクトリツリー中に適当な名前を移動先となるディレクトリを作成し、各種のディレクトリ情報を移動元のそれに設定する。ロック情報も移動元と同一とする(ステップ2503)。

【0168】(3) ステップ2503の処理でのディレクトリ作成処理が成功したか否かを確認する。ディレクトリ作成に失敗していた場合、ディレクトリ実体の移動処理失敗として処理を終了する(ステップ2504、2521)。

【0169】(4) ステップ2504の確認で、ディレクトリ作成に成功していた場合、移動元のディレクトリ内に存在する名前エントリのリストL1を取得する。但し、このとき、自分自身とファイルシステム内での親ディレクトリ、外部リンクによる親ディレクトリに関するエントリとは含めない。その後、前述で取得した名前エントリのリストL1の先頭を確認対象として選択して設定する。なお、名前エントリのリストL1が空の場合、確認対象なしとする(ステップ2505、2506)。

【0170】(5) 次に、ステップ2506で選択した確認対象が存在するか否かを確認し、存在した場合、確認対象の実体の種類を確認する。このとき、確認対象が外部リンクである場合、リンク先の実体により種類を確認する。この確認で、実体がファイル、あるいは、ディレクトリの場合、確認対象であるファイルあるいはディレクトリの名前の変更(rename)処理により、移動元ディレクトリから移動先ディレクトリへの名前の変更で移動を実現する。この名前変更処理は、必ず外部リンクを考慮する必要がある処理となるが、前述の名前変更処理により処理が可能である(ステップ2507~2509)。

【0171】(6) ステップ2508での確認で、確認対象の実体の種類がファイル、あるいは、ディレクトリ以外のものである場合、それに対応する移動処理を実行する。このとき、各種情報を全て移動先にコピーする処理を実施する(ステップ2510)。

【0172】(7) ステップ2509、2510の処理後、ステップ2509あるいはステップ2510の処理が成功したか否かを確認する。この確認で、ステップ2509または2510での処理が成功していた場合、名前エントリのリストL1の現確認対象の次のエントリを確認対象として選択して設定する。なお、名前エントリの次エントリが存在しない場合には、確認対象なしとする。この選択した確認対象について、ステップ2507からの処理に戻って処理を続ける(ステップ2512)。

【0173】(8) ステップ2507の確認で、選択した確認対象が存在しなかった場合、自分自身とファイルシステム内での親ディレクトリ、外部リンクによる親ディレクトリに関するエントリ以外の全ての名前エントリに関する処理が完了していることを意味する。このため、ここでの処理では、移動元のディレクトリと移動先のディレクトリとの名前とを変更することにより、名前空間において、移動先のディレクトリを参照するようにする。すなわち、名前変更処理で移動元のディレクトリを適当な名前に変更してバックアップし、その後、移動先として作成したディレクトリを移動元として指定された名前に変更する(ステップ2514)。

【0174】(9) 次に、ステップ2514での処理が成功したか否かを確認し、成功していた場合、移動対象のディレクトリに対して移動処理中にかけられていたロックを解除し、ステップ2514でバックアップとして別の名前に付け替えられた移動元のディレクトリを削除する。このとき、ファイル、ディレクトリ以外のエントリが残っている可能性があるため、まずこれを削除した後にディレクトリの削除を行う。これらの処理の後、ディレクトリ実体の移動処理成功としてここでの処理を終了する(ステップ2515~2517、2520)。

【0175】(10) ステップ2515での確認で、ステップ2514での処理が失敗していた場合、すなわち、ディレクトリ実体の移動処理に失敗した場合、まず、移動先ディレクトリに指定されたディレクトリに名前が変更されたディレクトリやファイルを移動元のディレクトリに戻す処理を行う。その後、コピーされたその他のエントリを削除し、移動先として作成されたディレクトリを削除する(ステップ2518)。

【0176】(11) 次に、移動対象のディレクトリに対して移動処理中にかけられていたロックを解除し、ディレクトリ実体の移動処理失敗として処理を終了する(ステップ2519、2521)。

【0177】図22はファイル実体を別のファイルサー

バに上に存在する他のディレクトリツリーへ移動する処理を説明するフローチャートであり、以下、これについて説明する。この処理において、実体を移動するファイルは、名前空間全体によるパス名で指定されるものとする。移動先は、ファイルサーバとそのファイルサーバ内のディレクトリツリーとが指定されるとする。

【0178】(1) ファイル実体の移動処理が開始されると、移動対象に指定されたファイル実体に通常アクセスに対するロックを設定する。このロックは、移動処理中に中身の変更が行われることがないようにするためである。そして、このロックは、実体を管理するデータの中にロックを管理するフラグを設定し、このフラグがセットされている間、アクセスを行わないようにすることにより実現する。但し、このロックが設定されていても、移動処理に関するアクセスは可能である(ステップ2601、2602)。

【0179】(2) 次に、移動の指示に使われた名前が保持されたディレクトリの中に移動ファイル実体への名前エントリが幾つあるか確認する。このとき、移動指示に使われた名前が外部リンクであった場合、リンク先ファイルサーバにおける外部リンクの被参照を管理するディレクトリにおいて、移動ファイル実体への名前エントリが幾つあるか確認する(ステップ2603)。

【0180】(3) 次に、ステップ2603で取得した名前指示ディレクトリ内での移動ファイル実体への名前エントリ数が、そのファイル実体の管理情報中に記録されている名前数と同一か否かを確認する。もし、同一であれば、指定名前を保持するディレクトリ内に存在する指定されたファイル実体へのパス名を全て取得する。但し、移動指示に使われた名前が外部リンクであった場合、リンク先ファイルサーバにおける外部リンクの被参照を管理するディレクトリにおける移動ファイル実体へのパス名を全て取得する(ステップ2604、2605)。

【0181】(4) ステップ2604の確認で、名前エントリ数が、そのファイル実体の管理情報中に記録されている名前数と一致しなかった場合、ファイル実体が記憶されているディレクトリツリー内全体において移動指定がなされたファイル実体を示す名前エントリを探す。このとき、名前エントリが見つかった場合には、そのときのパス名を全て保存する。なお、この処理は、ディレクトリツリー全体の検索を要するため、処理に時間がかかる可能性がある(ステップ2606)。

【0182】(5) ステップ2606の処理で取得したファイル実体が記憶されているディレクトリツリー内での同一ファイル実体への名前エントリ数が、そのファイル実体の管理情報中に記録されている名前数と同一か否かを確認する(ステップ2607)。

【0183】(6) ステップ2607の確認で、名前エントリ数が、そのファイル実体の管理情報中に記録され

ている名前数と同一であった場合、または、ステップ 2605 の処理の後、ステップ 2605、ステップ 2606 の処理で取得したパス名のうち、外部リンクの被参照を管理するディレクトリ内の移動ファイル実体へのパス名全てについて、そのリンクを参照するファイルサーバとそのファイルサーバに記憶されているディレクトリツリーと、その名前エントリへのパス名とを取得する。被参照を管理する名前の中には、名前空間ファイルサーバ利用管理情報 401 におけるツリー ID と、ツリー ID で識別されるディレクトリツリーを保持するファイルシステムにおける外部リンク元の名前エントリを保持するディレクトリの識別子とを保持しているため、そのリンク参照元ファイルサーバに対してリンク元のパス名の解決を依頼することができる（ステップ 2608）。

【0184】（7）次に、ステップ 2608 の処理における外部リンクの被参照名に関するリンク元パス名の取得処理が成功したか否かを確認する。もし、成功していた場合、ファイル実体を移動するために指定ファイルサーバの指定ディレクトリツリーにコピーする処理を実行する。コピー名は、適当なものを設定する。コピー完了後、ファイル実体を管理する情報もコピーもとのそれと同じものに設定する（ステップ 2609、2610）。

【0185】（8）ステップ 2610 でのコピー処理が成功したか否かを確認を行い、成功していた場合、移動ファイル実体を参照する名前について、移動前データを参照する名前を別の名前にバックアップし、その後、元の名前でコピーデータに対するリンクを作成する。移動元ファイル実体が存在するディレクトリツリーと同じディレクトリツリーに存在する移動ファイル実体を参照する名前のパス名は、ステップ 2605 または 2606 の処理で取得されており、移動元ファイル実体に対して外部リンクで参照する名前のパス名についてはステップ 2608 で全て取得されている（ステップ 2611、2612）。

【0186】（9）ステップ 2612 の処理が全て成功したか否かを確認を行い、成功していた場合、ファイル実体の移動処理が完了したことを意味するので、後始末の処理を実施する。すなわち、ステップ 2602 で設定した移動対象に指定されたファイル実体に通常アクセスに対するロックを解除し、ステップ 2612 で別の名前にバックアップしておいた名前を全て削除する。これにより移動前ファイル実体が削除される。さらに、ステップ 2610 でコピー作成時に利用した名前も不要となったので削除する。これらの処理後、ファイル実体の移動処理成功として処理を終了する（ステップ 2613～2615、2619）。

【0187】（10）ステップ 2613 での確認で、ステップ 2612 の処理が成功していなかった場合、処理失敗による復元処理として、すでに移動後のファイル実体を参照するように変更されたものについて、新規に作り

直した名前を削除して、バックアップしていたものを元に戻す処理を行う（ステップ 2616）。

【0188】（11）ステップ 2616 の処理の後、または、ステップ 2611 でのデータコピー処理が成功したか否かの確認で、データコピーに失敗していた場合、処理失敗による復元処理であるコピーデータの削除処理を行う（ステップ 2617）。

【0189】（12）ステップ 2617 の処理後、または、ステップ 2607 の確認で、名前エントリ数が、そのファイル実体の管理情報中に記録されている名前数と不一致であった場合、あるいは、ステップ 2609 の確認で、外部リンクの被参照名に関するリンク元パス名の取得処理に失敗していた場合、ステップ 2602 で設定した移動対象に指定されたファイル実体に対して、通常アクセスに対するロックを解除し、その後、ファイル実体の移動処理失敗として処理を終了する（ステップ 2618、2620）。

【0190】ファイル共有クライアント 112c、112d は、前述した処理において、ファイルやディレクトリの実体移動が行われた後、この移動を知る術がないので、昔の位置に対してアクセスを行うことになる。その場合、ファイル共有クライアント 112c、112d には、指定されたファイルやディレクトリが未存在のエラーが返されることになる。そこで、ファイル共有クライアント 112c、112d の中に存在するファイル共有外部アクセス実行部 146 は、ファイル未存在エラーが返却された際に、再度エラーが帰ってきたファイルやディレクトリへのパス名を評価し、その値を元に再度アクセスを試みる。

【0191】移動を行う際には、移動対象の決定方法が重要になる。移動対象の決定方法としては、以下のような方法を使用することができる。

【0192】1つの方法は、ある特定のディレクトリ以下に所属する全てのファイルとディレクトリとを一度に移動する方法である。この方法は、同じように利用されるデータを纏めて管理することが好ましいことを考慮した方法である。この移動の処理は、あるディレクトリを指定した後に、そのディレクトリ以下を根元から、あるいは、枝先から再帰的に移動処理を行うことにより実現することができる。この移動を行う際には、どの程度のデータ量が移動されるかを判断しておく必要がある。

【0193】また、別の方法は、ある特定の属性を保持するデータのみを分離して移動する方法である。例えば、ファイルシステムが最終アクセス時刻を管理している場合に、最終アクセス時刻が古く利用頻度が低いファイルを特定のファイルサーバ上に存在するディレクトリツリーに移動するようにする方法である。

【0194】この最終アクセス時刻による移動対象決定方法は、領域不足時に自動的にファイルを移動するために利用することができる。この方法は、移動が必要にな

ったときに、最終アクセス時刻が古いファイルを検索するために時間がかかるので、定期的にディレクトリツリー内を検索して、最終アクセス時刻が古いもののリストを作成しておき、実際に移動させる必要が生じた場合に、最終アクセス時刻が古いもののリストを参照して古い順に移動対象の候補を決め、その後に再度最終アクセス時刻を確認して本当に最近利用されていないことを確認した後にそのファイルを移動させるように実行される。

【0195】本発明の実施形態は、複数ファイルサーバのファイルシステム上の一部領域であるディレクトリツリー156a、156b、156c、156dをまとめて1つの名前空間として見せるために、ファイルシステム管理部142が他ファイルサーバ上のファイルシステムへの外部リンクの機能を保持しているが、本発明は、前述の外部リンク機能の代わりに別途名前空間を管理するサーバ機能を用意し、そこで名前の管理を行うことも可能である。この場合にも、前述した実施形態の場合と同様の方法で、各ファイルシステムにおいて、ファイルシステムを幾つかのディレクトリツリーに分割し、それぞれ独立にファイル利用量・ファイル作成数の制限を設ける機能を保持し、それを連携して管理することにより1つの名前空間全体としてファイル利用量・ファイル作成数の制限がかけられているように運用することができ。

【0196】前述した本発明の第1の実施形態によれば、他ファイルサーバ上にファイルとして領域を割り当て、そこをファイルサーバがローカルに利用している外部記憶装置と等価に利用することが可能となり、これにより、ファイルサーバが利用するファイルシステムの機能に何ら影響を与えることなく、ファイルサーバが他ファイルサーバ上の領域を利用することができる。

【0197】また、本発明の第2の実施形態によれば、各ファイルサーバのファイルシステムにおいて、他のファイルシステム上のディレクトリ実体やファイル実体に対するリンク情報を保持してそれを適切に管理することができるので、複数ファイルサーバに跨った1つの名前空間を形成し、その中でファイル実体に対して等価な複数の名前を付加する機能を保持することができる。さらに、第2の実施形態によれば、ファイルサーバ内のファイルシステムを幾つかの名前管理の空間に分割し、その集合として名前空間を提供し、分割された名前管理の空間に、それぞれ独立にファイル利用量・ファイル作成数の制限を設ける機能を保持させ、この制限値の管理を1つの名前空間を提供する複数ファイルサーバ間で連携して行うことができるので、名前空間全体を1つの管理単位としてファイル利用総量・ファイル作成数の制限が行われているように動作させることが可能となる。

【0198】

【発明の効果】以上説明したように本発明によれば、フ

ァイルサーバが提供する名前空間やファイル利用量・ファイル作成数の制限とファイル実体に対して等価な複数の名前を付加する機能を保持しながら、あるファイルサーバが他ファイルサーバ上の領域を利用することを可能にすることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態によるファイルサーバを含む情報処理システムの構成を示すブロック図である。

10 【図2】図1の論理デバイス管理部保持するマッピング管理情報について説明する図である。

【図3】ファイルサーバでファイル共有関係の処理要求を受け取った後の処理動作を説明するフローチャートである。

【図4】他のファイルサーバ上に存在するファイルを新規作成し、あるいは、既存ファイルを拡張して利用することによりファイルサーバがファイルシステムの領域を拡張する処理動作を説明するフローチャートである。

20 【図5】他のファイルサーバ上に存在するファイルサーバの領域拡張用の外部ファイルの大きさを縮小する処理動作を説明するフローチャートである。

【図6】ファイルサーバが他のファイルサーバ上に割り当てている領域拡張用の外部ファイルをさらに第3のファイルサーバに移動させる処理動作を説明するフローチャートである。

【図7】本発明の第2の実施形態によるファイルサーバを含む情報処理システムの構成を示すブロック図である。

30 【図8】図7の論理デバイス管理部が保持するマッピング管理情報について説明する図である。

【図9】主ツリー管理情報の構成を説明する図である。

【図10】名前空間ファイルサーバ利用管理情報の構成を説明する図である。

【図11】名前空間ユーザ・グループ利用量管理情報の構成を説明する図である。

【図12】ディレクトリツリー内ユーザ・グループ利用量管理情報の構成を説明する図である。

【図13】副ツリー管理情報の構成を説明する図である。

40 【図14】ディレクトリツリー利用量管理情報の構成を説明する図である。

【図15】ディレクトリツリー全体のファイル利用量の割り当て制約にかかり、ファイル利用量が不足した場合の制約回避処理の動作を説明するフローチャートである。

【図16】ディレクトリツリー内のユーザやグループへのファイル利用量の割り当て制約にかかり、ファイル利用量が不足した場合の制約回避処理の動作を説明するフローチャートである。

50 【図17】外部リンク機能の外部リンク先おける被参照

名の決定方法について説明する図である。

【図 18】ディレクトリ内の外部リンクである名前エンタリを削除する場合の処理動作を説明するフローチャートである。

【図 19】あるファイル実体に対して等価に扱うことができる名前を新規に作成する場合に、ファイル実体が存在するディレクトリツリーと名前を新規に作成するディレクトリが所属するディレクトリツリーとが異なるサーバのファイルシステムに存在するときの外部エンタリ作成処理の動作を説明するフローチャートである。

【図 20】ファイルあるいはディレクトリの名前を変更する場合に、変更前名前のエンタリが外部リンクである、あるいは、変更前の名前エンタリを含むディレクトリが所属するディレクトリツリーと変更後の名前エンタリを含むディレクトリが所属するディレクトリツリーが異なるファイルサーバのファイルシステム上に存在する場合の名前変更処理の動作を説明するフローチャートである。

【図 21】ディレクトリ実体を別のファイルサーバの上に存在する他のディレクトリツリーへ移動する処理を説明するフローチャートである。

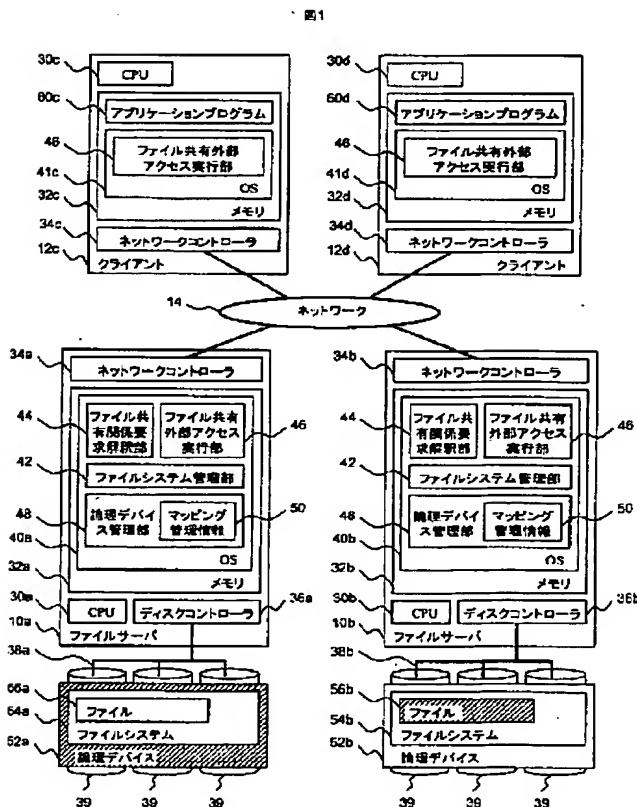
【図 22】ファイル実体を別のファイルサーバの上に存

在する他のディレクトリツリーへ移動する処理を説明するフローチャートである。

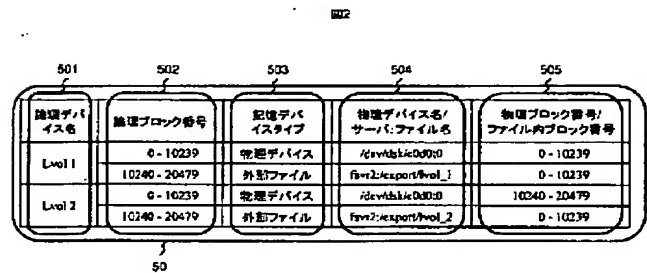
【符号の説明】

- 10 a、10 b ファイルサーバ 10 a
- 12 c、12 d ファイル共有クライアント
- 14 ネットワーク
- 30 a～30 d 中央処理装置 (CPU)
- 32 a～32 d メモリ
- 34 a～34 d ネットワークコントローラ
- 36 a、36 b ディスクコントローラ
- 38 a、38 b 外部記憶装置接続媒体
- 39 外部記憶装置
- 40 a、40 b、41 c、41 d オペレーティングシステム (OS)
- 42 ファイルシステム管理部
- 44 ファイル共有関係要求解釈部
- 46 ファイル共有外部アクセス実行部
- 48 論理デバイス管理部
- 52 a、52 b 論理デバイス
- 54 a、54 b ファイルシステム
- 56 a、56 b ファイル

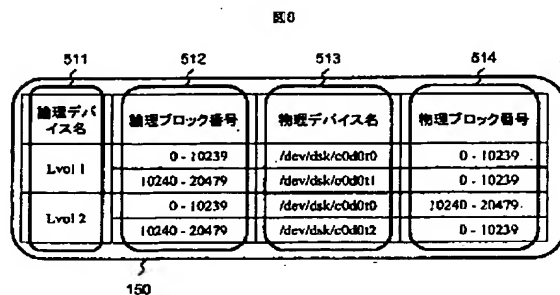
【図 1】



【図 2】

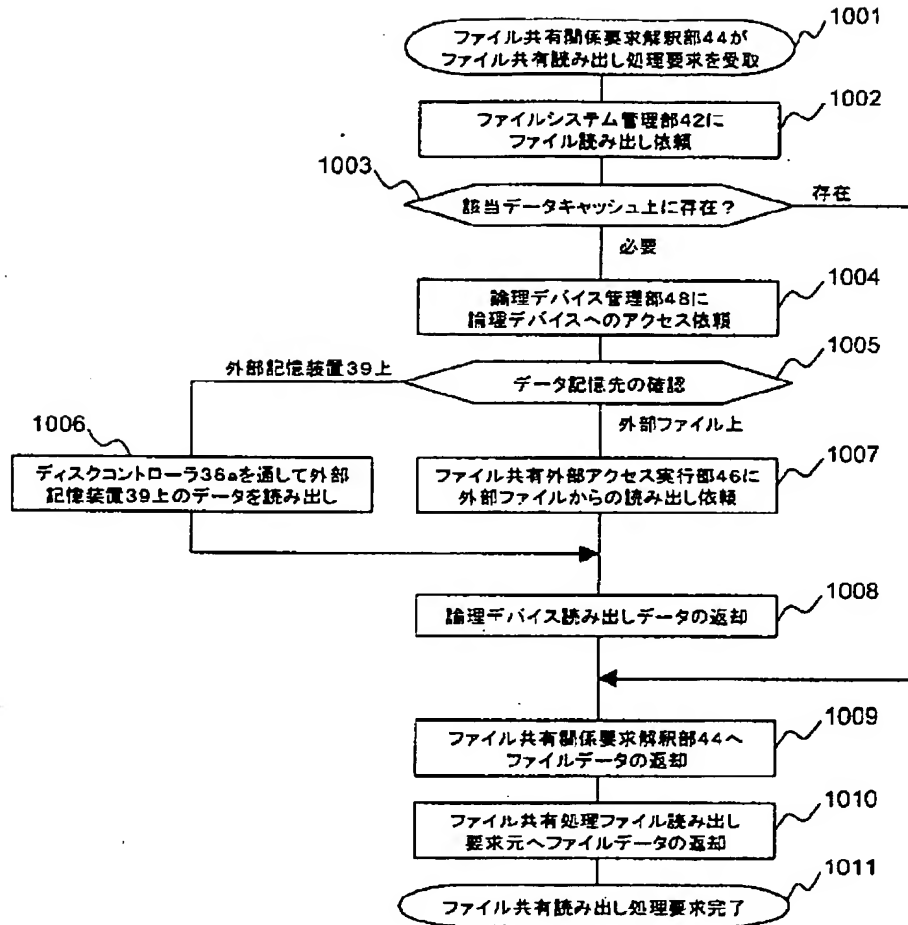


【図 8】



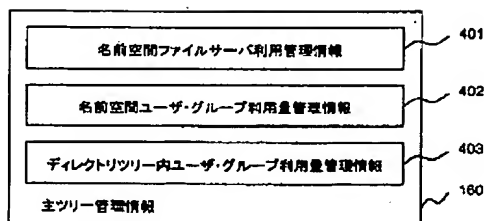
【図3】

図3



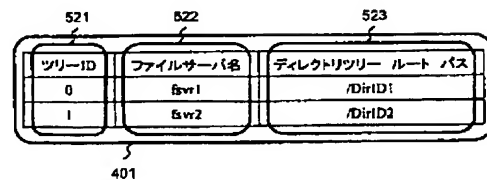
【図9】

図9



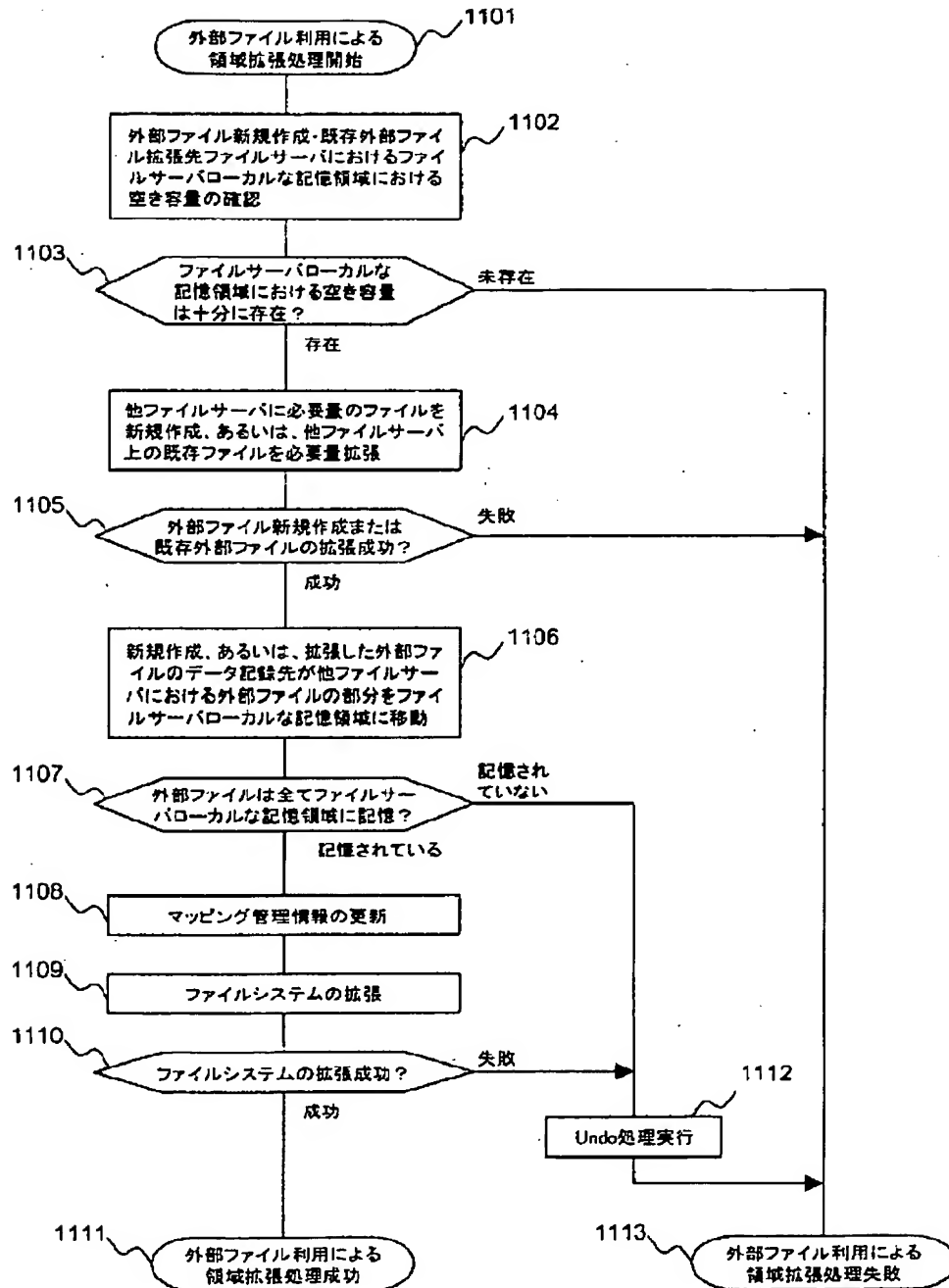
【図10】

図10



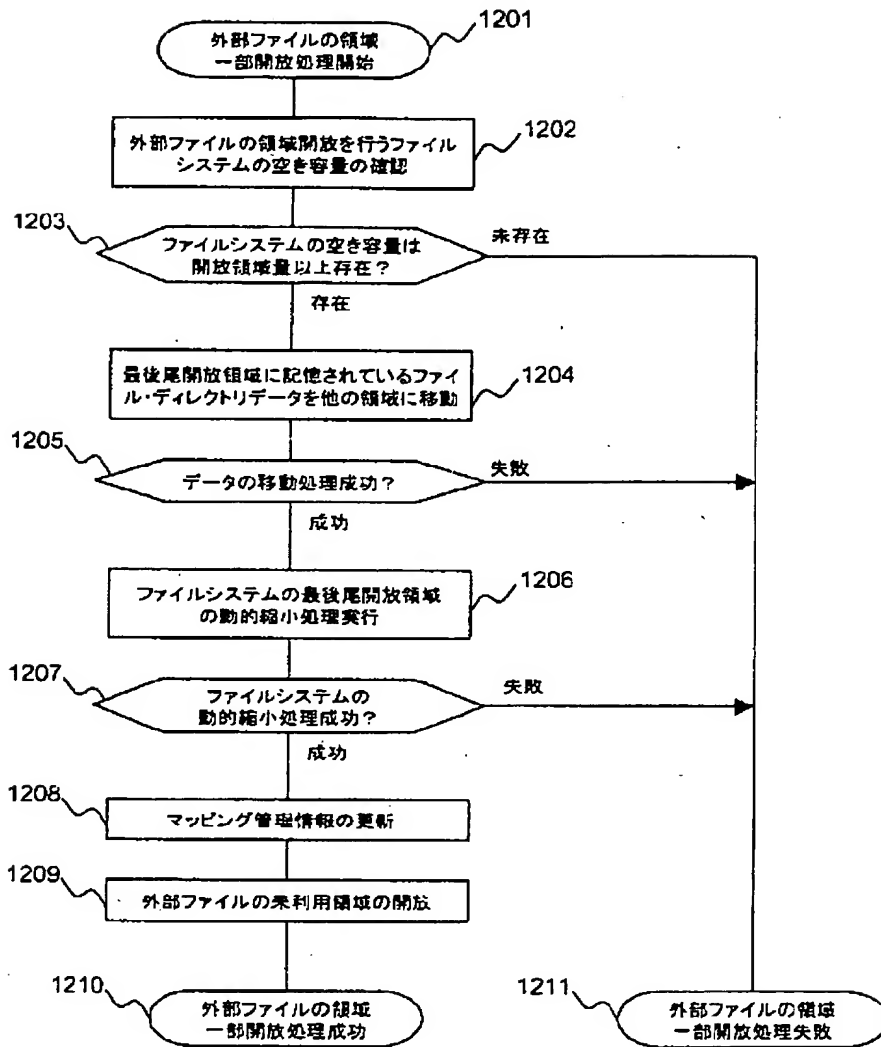
【図4】

図4



【図 5】

図5



【図 11】

図11

531	532	533	534	535	536
タイプ	ID	許可最大 ファイル利用量	許可最大 ファイル作成数	割り当て ファイル利用量	割り当て ファイル作成数
全体	-	10,000,000 kB	5,000,000	6,000,000 kB	3,000,000
User ID	12345	200,000 kB	100,000	200,000 kB	100,000
User ID	7872	100,000 kB	100,000	80,000 kB	80,000
Group ID	101	1,000,000 kB	500,000	700,000 kB	400,000
Group ID	3010	1,500,000 kB	750,000	1,500,000 kB	750,000

402

【図 12】

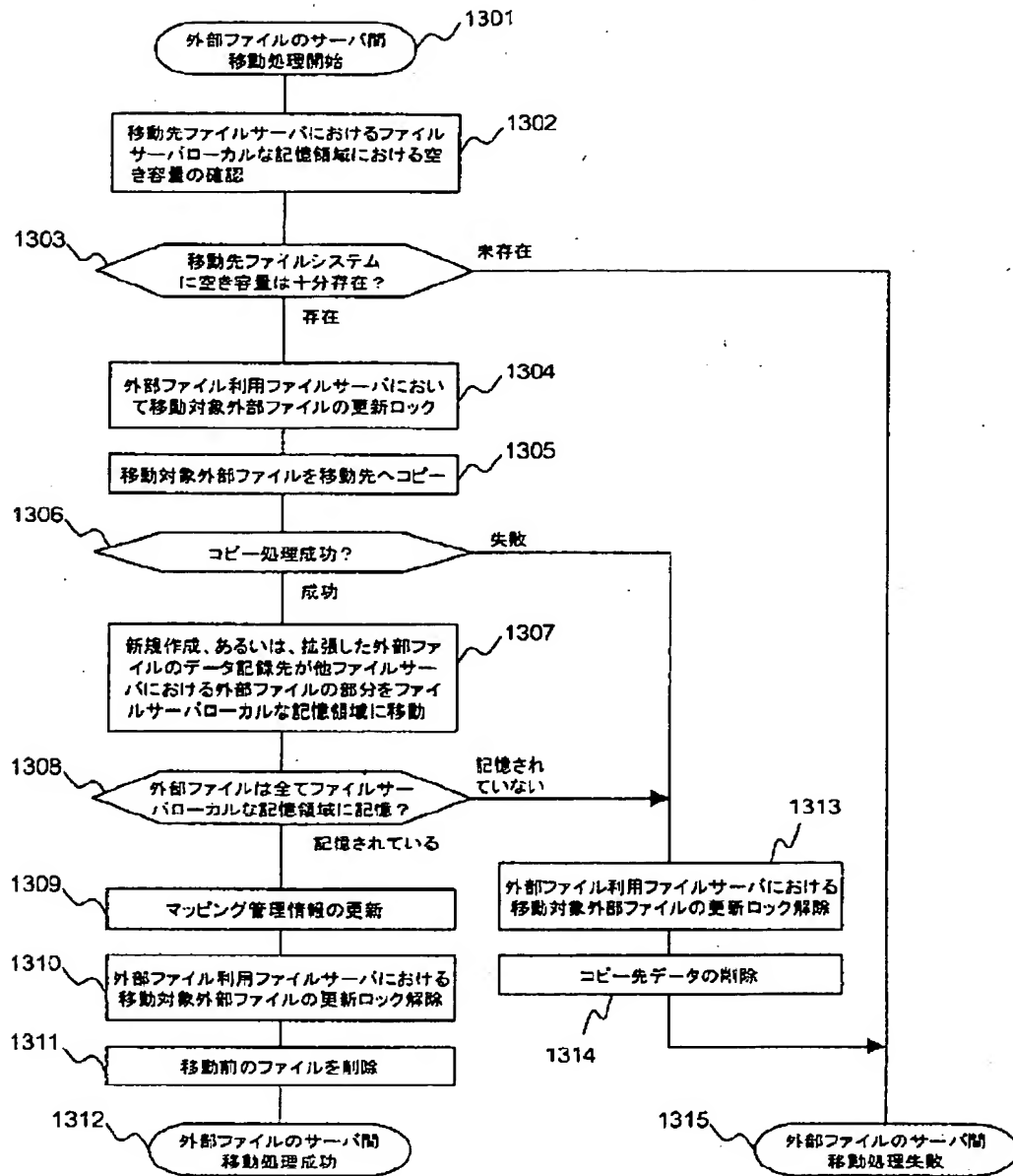
図12

541	542	543	544	545	546
タイプ	ID	割り当て ファイル利用量	割り当て ファイル作成数	現ファイル 利用量	現ファイル 作成数
全体	-	500,000 kB	500,000	200,000 kB	100,000
User ID	12345	200,000 kB	100,000	200,000 kB	100,000
User ID	7872	0 kB	0	0 kB	0
Group ID	101	0 kB	0	0 kB	0
Group ID	3010	300,000 kB	150,000	100,000 kB	50,000

403

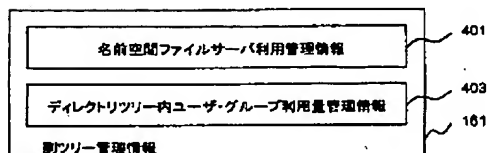
【図6】

図6

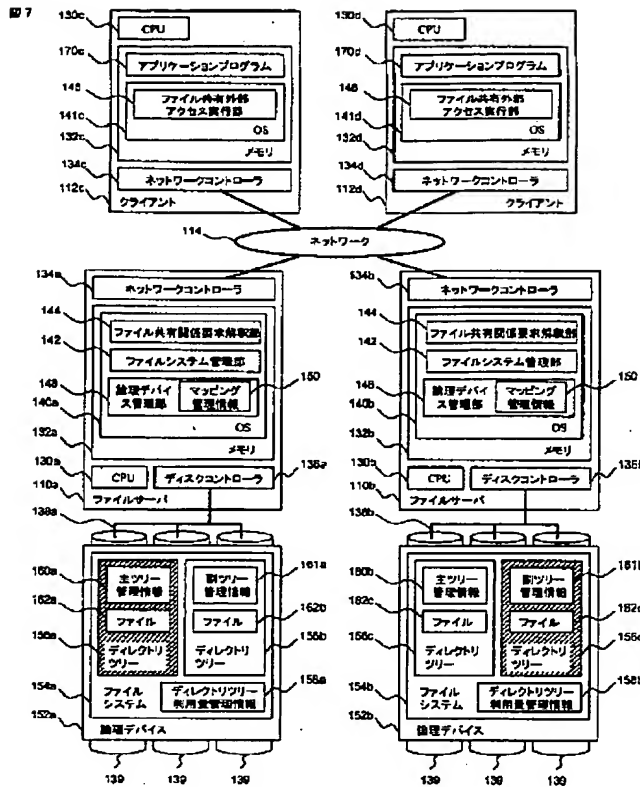


【図13】

図13

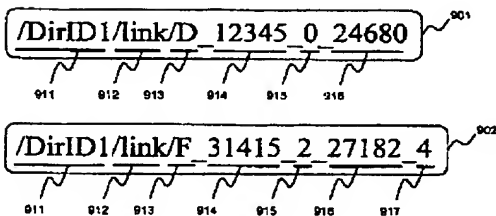


【図7】

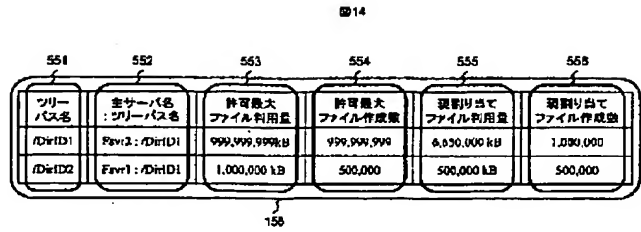


【図17】

図17

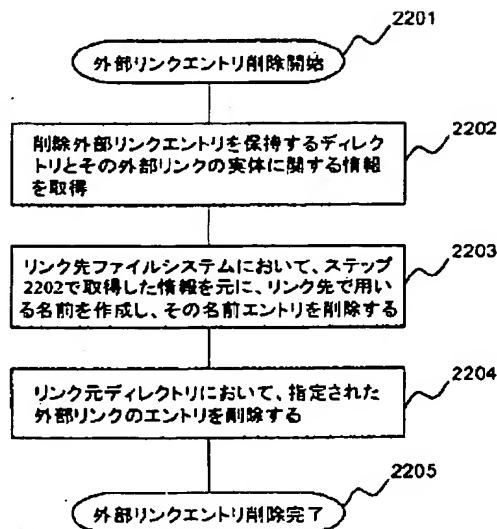


【図14】



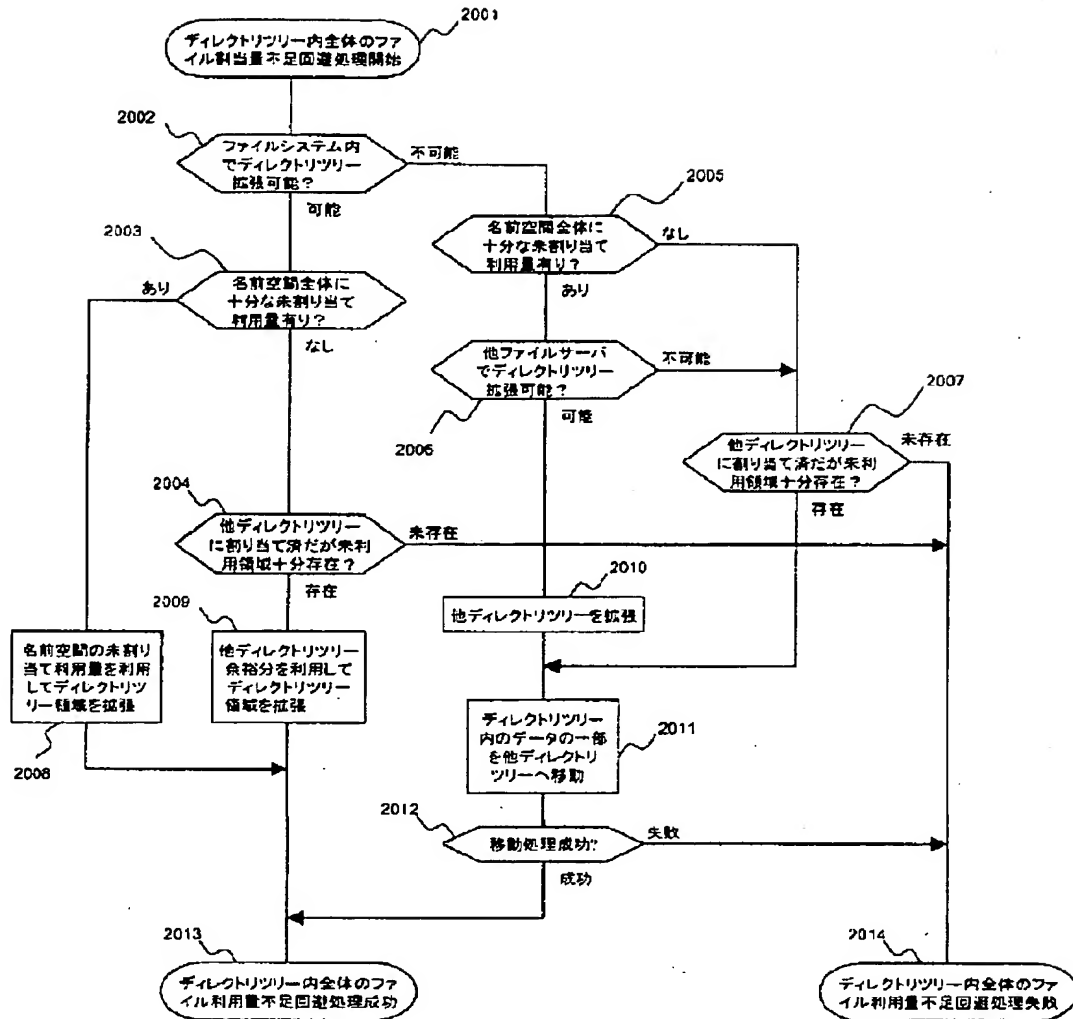
【図18】

図18



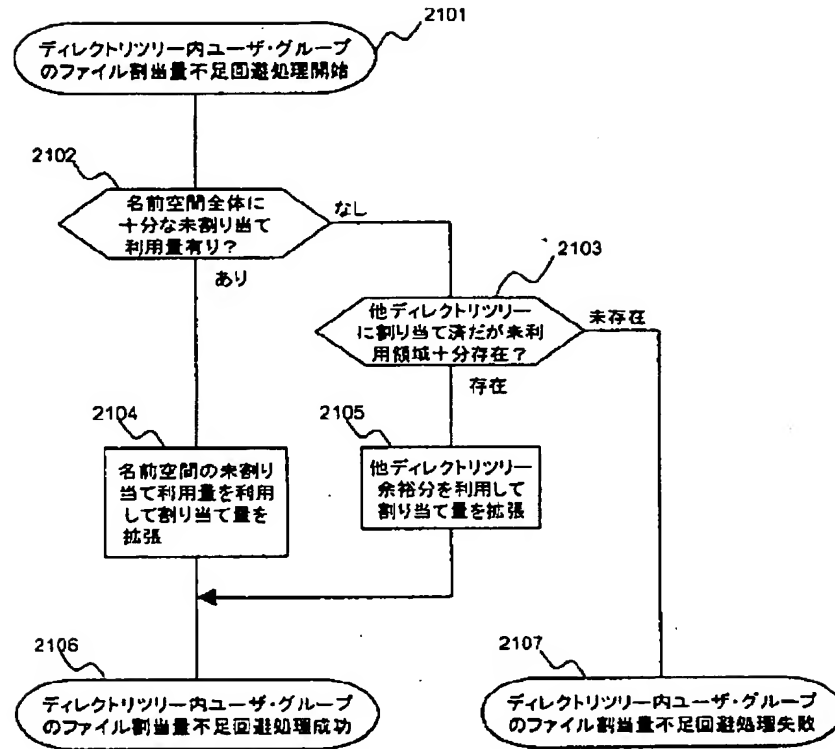
【図15】

図15



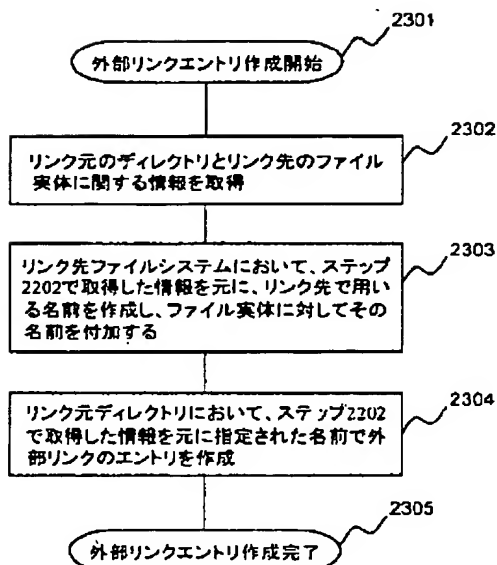
【図16】

図16



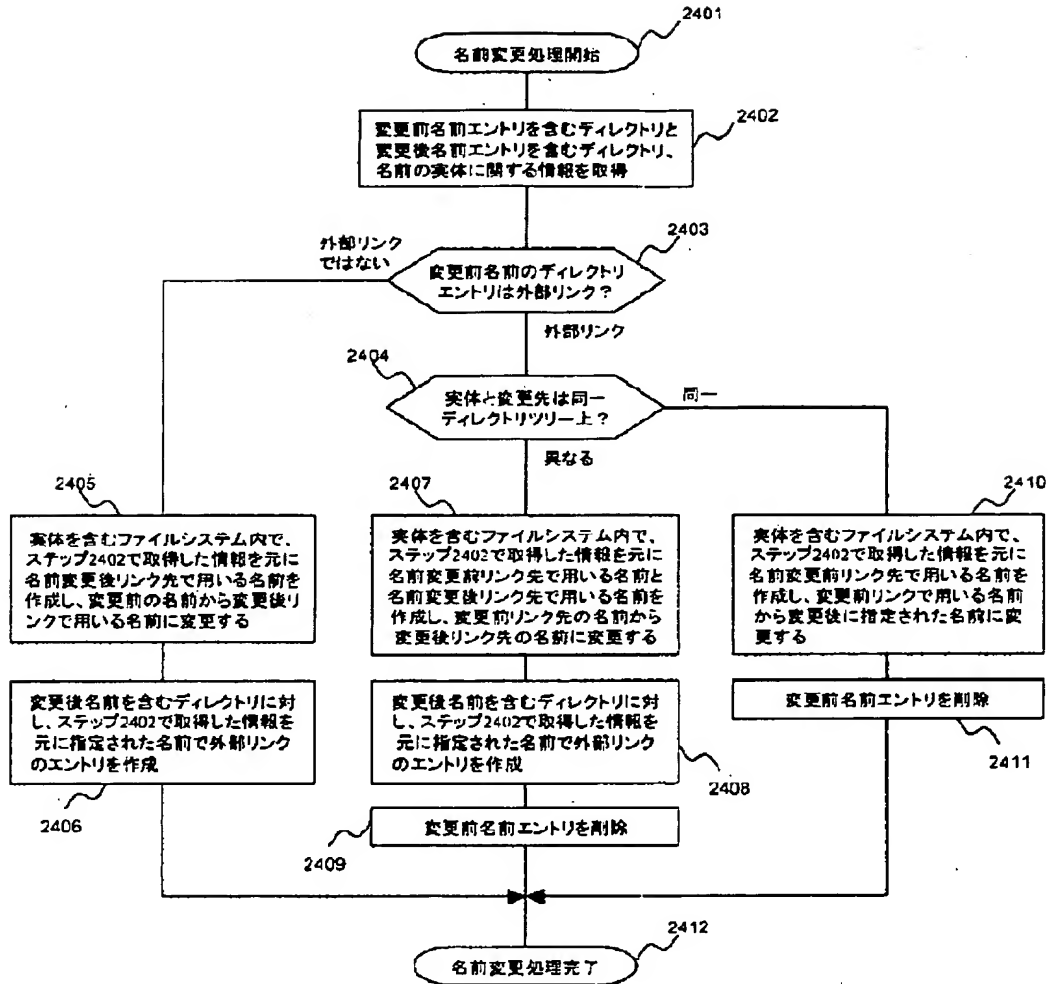
【図19】

図19



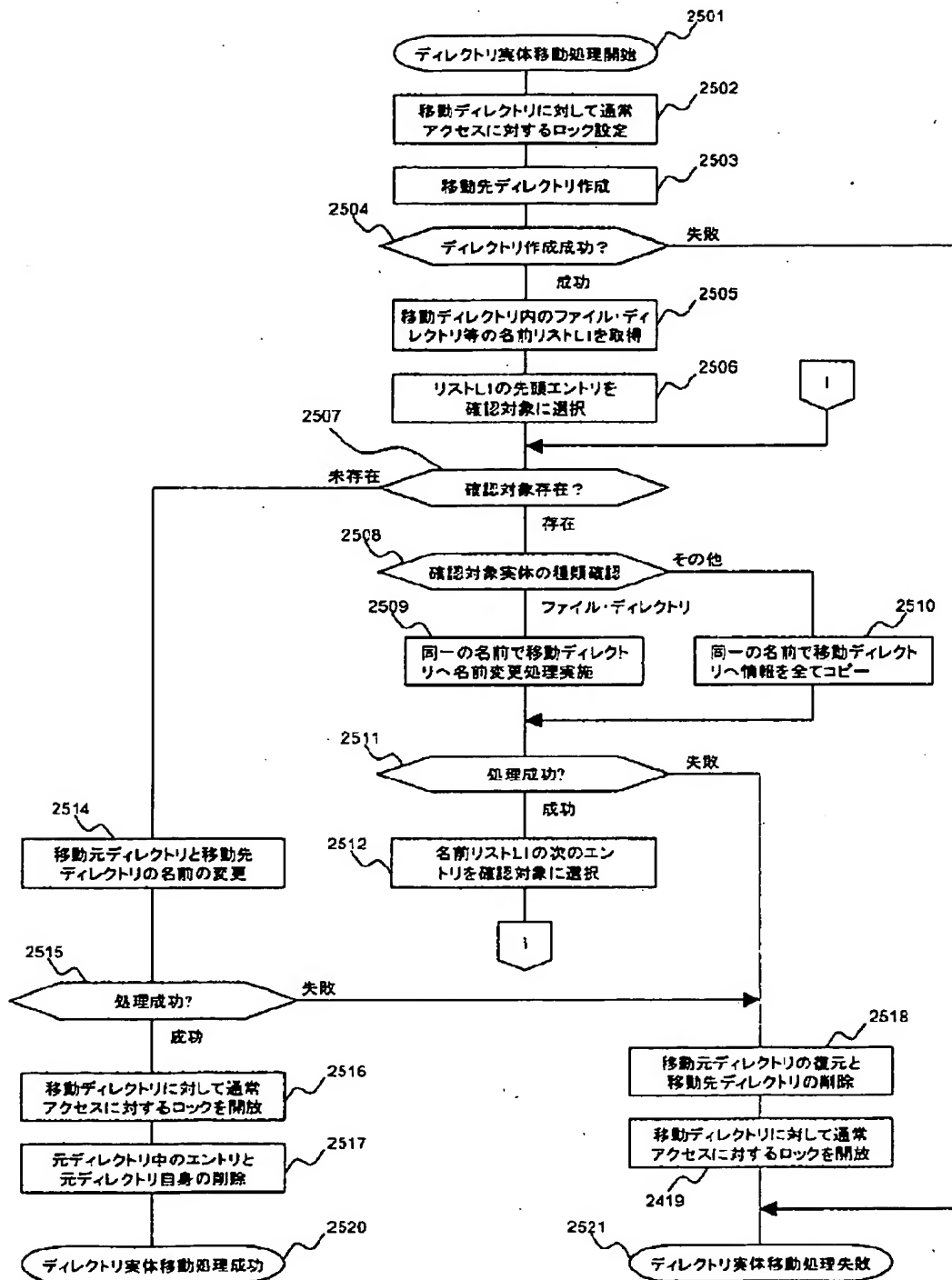
【図20】

図20



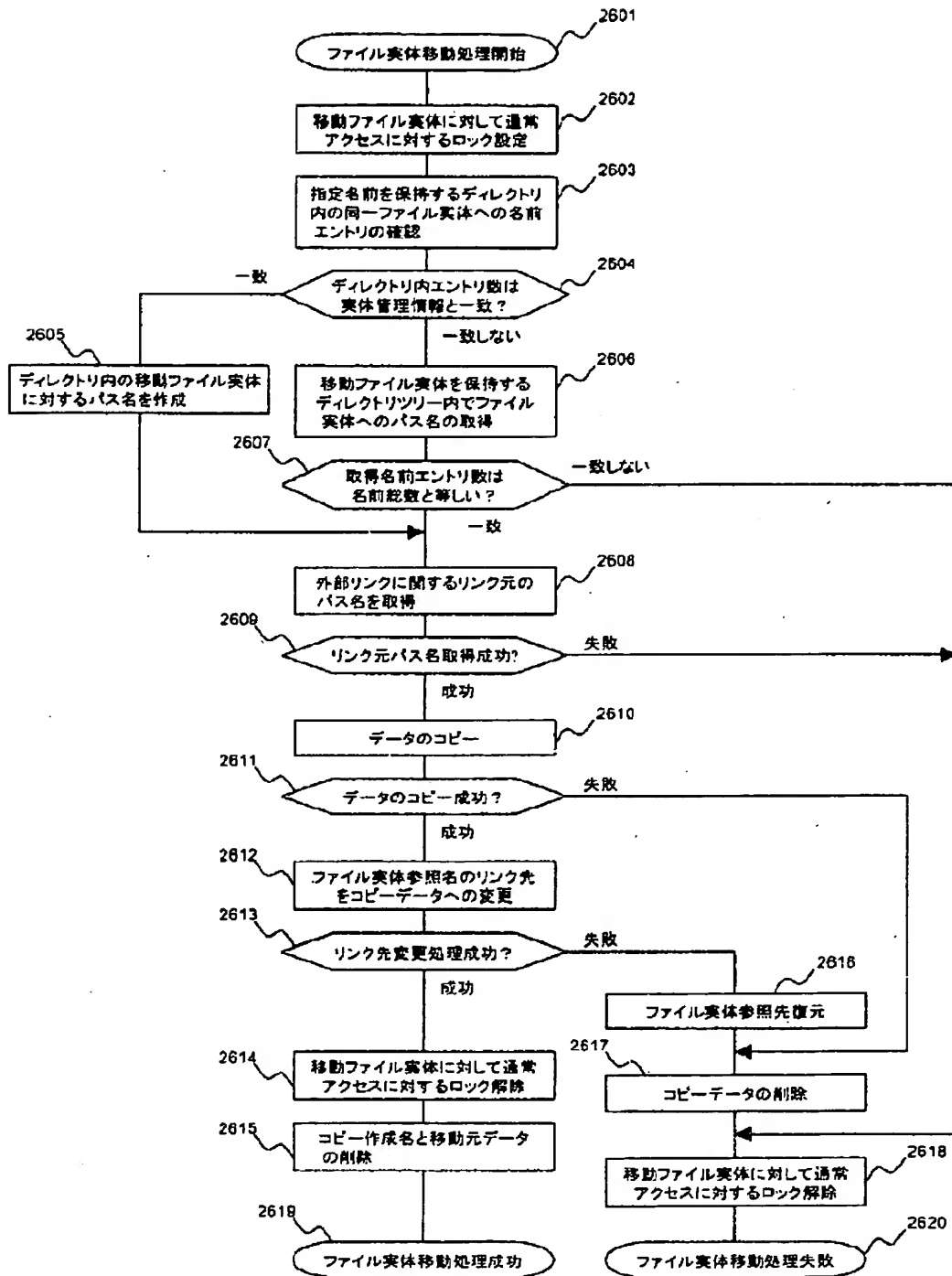
【図21】

図21



【図22】

図22



フロントページの続き

(72) 発明者 北村 学

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 荒井 弘治

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージ事業部内

Fターム(参考) 5B082 CA01 CA16 CA17 EA09